CONTEXTUAL HUMAN TRAJECTORY FORECASTING WITHIN INDOOR ENVIRONMENTS AND ITS APPLICATIONS

A Dissertation

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Pranav Mantini

December 2015

CONTEXTUAL HUMAN TRAJECTORY FORECASTING WITHIN INDOOR ENVIRONMENTS AND ITS APPLICATIONS

Pranav Mantini

APPROVED:

Shishir Shah, Chairman
Dept. of Computer Science

Edgar Gabriel
Dept. of Computer Science

Christoph Eick
Dept. of Computer Science

Guoning Chen
Dept. of Computer Science

Saurabh Prasad
Dept. of Electrical and Computer Engineering

Dean, College of Natural Sciences and Mathematics

Acknowledgements

Foremost, I would like to thank my advisor Dr. Shishir K. Shah for his patience, motivation, support and immense knowledge. I believe that I have been extremely fortunate in having him as my advisor. He provided me the freedom to explore and discover the research that excited me the most and supported me through my choices. I cannot imagine having a better mentor and advisor for my Ph. D. studies.

Besides my advisor, I'd like to thank the members of my committee Dr. Edgar Gabriel, Dr. Guoning Chen, Dr. Christoph Eick and Dr. Saurabh Prasad for their insight and invaluable suggestions in making this thesis better.

This would not have been possible without the unconditional love and support of my family. I'd like thank my mother Dr. N. Sailaja and my father M. P. Naidu for their utmost patience and my brother Pradosh for his continued support through my graduate studies. My mother has always been my role model, her perseverance and endeavor to pursue what the heart desires is inspiring. I thank her for inspiring me and supporting me through every step in my life. I would also like to thank Dr. Subba Rao Ghanta for his guidance and my aunt and uncle Uday and Aruna for their support.

Next, I'd like to thank my fellow members of Quantitative Imaging Laboratory. Chintan, Apurva, Khai, Xu, Xuqing, Can, Arko, Qazaleh, Ilyes, Adrian, David and others for their precious advise, support and thought provoking conversation through the years.

Next, I'd like to thank my friends Charu, Malerie, Paul, Charan, Kinjal, Joseph, Aura, Arshad, Radhika, Arun, Subash, Meenakshi, Nikhil, Imran and others for

being part of my academic and social life and making my graduate studies memorable and enjoyable. I am also grateful for having two adorable pets Anjali and Tanner who made me smile every time I opened the door to me house.

Finally, I'd like to thank my friends at Pratham and my running club for their support and the good time they provided me with.

Dedicated to my parents.

CONTEXTUAL HUMAN TRAJECTORY FORECASTING WITHIN INDOOR ENVIRONMENTS AND ITS APPLICATIONS

An Abstract of a Dissertation

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Pranav Mantini

December 2015

Abstract

A human trajectory is the likely path a human subject would take to get to a destination. Human trajectory forecasting algorithms try to estimate or predict this path. Such algorithms have wide applications in robotics, computer vision and video surveillance. Understanding the human behavior can provide useful information towards the design of these algorithms. Human trajectory forecasting algorithm is an interesting problem because the outcome is influenced by many factors, of which we believe that the destination, geometry of the environment, and the humans in it play a significant role. In addressing this problem, we propose a model to estimate the occupancy behavior of humans based on the geometry and behavioral norms. We also develop a trajectory forecasting algorithm that understands this occupancy and leverages it for trajectory forecasting in previously unseen geometries. The algorithm can be useful in a variety of applications. In this work, we show its utility in three applications, namely person re-identification, camera placement optimization, and human tracking. Experiments were performed with real world data and compared to state-of-the-art methods to assess the quality of the forecasting algorithm and the enhancement in the quality of the applications. Results obtained suggests a significant enhancement in the accuracy of trajectory forecasting and the computer vision applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Contextual Human Trajectory Forecasting

Given a human subject and their destination, trajectory forecasting deals with predicting or estimating the likely path a subject will take to reach the destination. Trajectory forecasting has a variety of applications. In robotics, it can be used for robot motion planning, in surveillance it can be used for predicting the future location of subjects and could also be used to improve the accuracy of computer vision algorithms for tracking, re-acquisition, etc. Networked cameras are widely used for monitoring human activity in public areas. Camera networks spanning from hundreds to thousands of cameras per network is a common occurrence in busy public locations like airports. Most of these cameras might have non-overlapping fields of view. A holistic automated surveillance system cannot infer a semantic understanding of the scenario without a model for linking the observed actions from individual cameras.

The surveillance system should have an understanding of the 3D geometry of the environment it is present in, along with an understanding of the relation between the cameras. Considerable effort is focused on automatic generation of 3D models for outdoor and indoor environments [12, 13, 51, 79, 14]. Furthermore, reasonable attempts have been made in understanding the camera topography [20, 63, 28, 90] for applications like tracking [50, 77] and re-identification [67, 60, 59]. In these cases, it is very essential to predict the trajectories of humans based on the geometry of the environment. For example consider a re-identification problem, where a human is observed in two different cameras in the same network. Estimate of the trajectory starting from the observation in the first camera to the destination in the second can impart an approximate spatial and temporal knowledge of the human's actions. This can assist in designing robust re-identification algorithms. Similarly, human motion and estimation of trajectories are critical in urban planning where the design of new public spaces and their geometries will be influenced by simulations of expected human occupancy and their movements [5, 40].

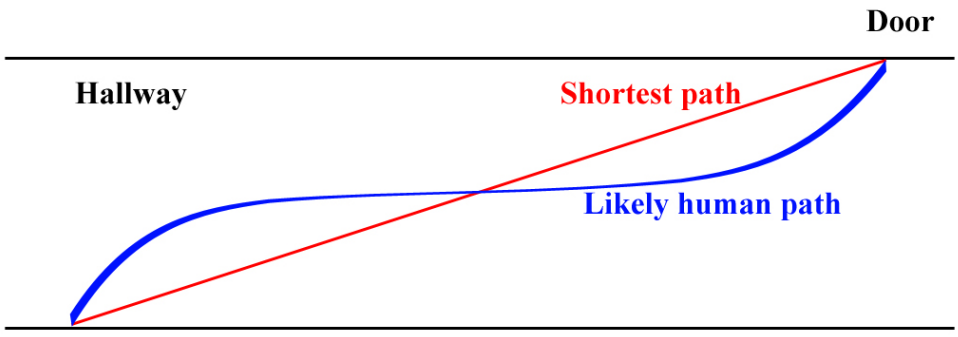


Figure 1.1: Shortest path vs. likely path

Human motion is influenced by a multitude of factors, many of which are driven by perception. It is well understood that 3D geometry and the physical world imposes specific constraints on human motion. In many cases, interaction of humans with the surrounding geometry and humans is not explicitly modeled. In general, we can agree that a trajectory followed by a human subject is motivated by the target destination, but it is not necessary that they would always take the shortest path defined by the geometry. Though the human subject's main motivation is to reach the destination in the shortest time, they tend to subconsciously follow some behavioral norms. For example, if a person is trying to reach a door that is on the left hand side at the far end of a hallway, they would walk in the center or close to the center of the hallway almost all the way until they get close to the door as shown in Figure 1.1. In this case, the shortest distance is to stay as close to the left wall of the hallway as it is physically possible, but we rarely see such behavior. This behavior, though at a subconscious level, is being influenced by the surrounding 3D geometry and behavioral norms. Our continuous interaction with different geometries in various environments over time may have led to the evolution of this behavior. This work is focused on indoor environments whose 3D geometry is available and performs human motion prediction while accounting for destination, 3D geometry and humans in the environment.

Destination is the motivating factor for human motion. Most people traverse with the objective of reaching a destination, usually within the shortest time possible while adhering to social norms imposed by the geometry and other humans.

Geometry of the environment (like the walls and doorways) and static objects

have an effect on the human motion. For example in a narrow hallway with walls on either side, majority of the humans prefer to walk at the center of the hallway as opposed to the edge. Furthermore, in a classroom, consider how one would navigate around tables and chairs or any other object to get to a seat.

Contextual Trajectory Forecasting (CTF) takes into account the above factors and their influence on human motion to estimate the likely path a human might take.

Humans in the environment are dynamic and the human motion of the subject is effected by other humans and vice versa. For example, consider how humans plan their motion to navigate around other humans while maintaining some socially acceptable distance.

1.2 Applications

Human motion prediction algorithms have a wide variety of application from robotics to construction planning. In this work, the applicability is demonstrated in three computer vision applications, namely person re-identification, camera network placement optimization, and human tracking.

1.2.1 Person Re-Identification

Person re-identification (re-ID) is the ability to associate the identity of a person observed at one time and location with the same subject when observed at a different time and location. Given the observation of an individual from different cameras,

over disparate time and location, automated re-identification algorithms deal with the task of associating the identity correctly of the individual across all observations. Re-ID is an everyday trivial task for human beings. Replicating such system is a confounding task because of various difficulties originating from low quality images, occlusion, changes in illumination, view, and pose across cameras [9]. Furthermore the lack of robust algorithms to infer the topology of the network and calibrate camera locations for leveraging contextual information complicates this process.

Networked cameras are widely used for monitoring human activity in public areas. Camera networks spanning from hundreds to thousands of cameras per network are a common occurrence in busy public locations like airports. These camera networks generate massive quantities of video data. At the time of need, manually fishing for a single human subject from the sea of data is a tedious and time-consuming task. Re-ID algorithms find a natural place in these scenarios. Given the videos from these networks, the task of re-ID performed by security personnels though tedious is still extremely reliable. To design a re-ID algorithm, we take motivation from how these security personnels might use a combination of appearance features along with contextual information to identify subjects in the videos. For example, if a person is observed in a particular video, the human performing re-ID will notice information such as the color of clothing, the direction of motion and their velocity (run or walk). The human performing re-ID has knowledge regarding the geometry of the environment, the topology of the camera network. This knowledge can be used to process the observed information to arrive at an estimate of the likely future time and location of the observed person. This estimate will allow the human to only

search for a small window of time in specific locations for a match. Hence, using information regarding the geometry of the environment can be of vital assistance in re-ID.

Re-ID involves feature matching to find an identity in the database with similar features. The features encompass information regarding appearance of the person like color and texture, or context of the scenario like the time and location of the human subject. CTF provides a future estimate of the likely time and spatial location of previously observed subjects. Embedding this information into traditional re-ID algorithm significantly boost their performance.

1.2.2 Camera Network Placement Optimization

Video surveillance is an integral part of many public areas such as airports, banks and train stations. The positioning and orientation of the cameras can play a significant role in enabling effective surveillance needs such as face detection, tracking, etc. The geographic distribution of cameras to enable effective surveillance can be scenario specific. For example, in a movie theater, it might be sufficient to deploy cameras at locations that exhibit dominant human activity, but at an airport, it may be imperative to deploy cameras to obtain a maximum visibility of observable space along with emphasis on areas with dominant human activity. Some common factors that should be taken into consideration while deploying cameras include visibility coverage and deployment costs.

Visibility coverage: In high security scenarios, the camera configuration should

be optimized such that a maximal coverage of the observable space in the infrastructure can be obtained along with added emphasis on areas with dominant human activity. In low-security scenarios, the camera configuration should at least guarantee the coverage of all the areas where dominant human activity would take place. The configuration can be made more effective by covering the most frequently used entry and exit points in the infrastructure. Furthermore, a camera configuration that maximizes the capture of specific pose of objects of interest (e.g., frontal image of the humans) with sufficient resolution is considered more effective.

Deployment cost: The configuration should guarantee the mentioned visibility coverage while deploying the least required number of cameras. Furthermore, having a minimal number of cameras has a significant impact on the available storage space with HD cameras becoming more prevalent and requiring higher storage space.

Designing a camera deployment configuration manually by taking into consideration the above factors can be extremely tedious and error prone. Automated camera network deployment techniques are essential for a cost effective and safe environment. In this application, we address the issue of obtaining effective surveillance by optimizing the deployment of cameras. In doing so, the multi-factorial issues of visibility coverage, deployment costs, preferred pose of objects of interest and resolution are considered. In this work, a camera configuration is considered to provide effective surveillance if the views across deployed cameras maximizes the following aspects while minimizing the total number of cameras.

- the observable space,

- the view of regions within the infrastructure where dominant activity is expected,
- the ability to capture the preferred pose of objects of interest (e.g., frontal pose of humans) and
- their image resolution (e.g., face).

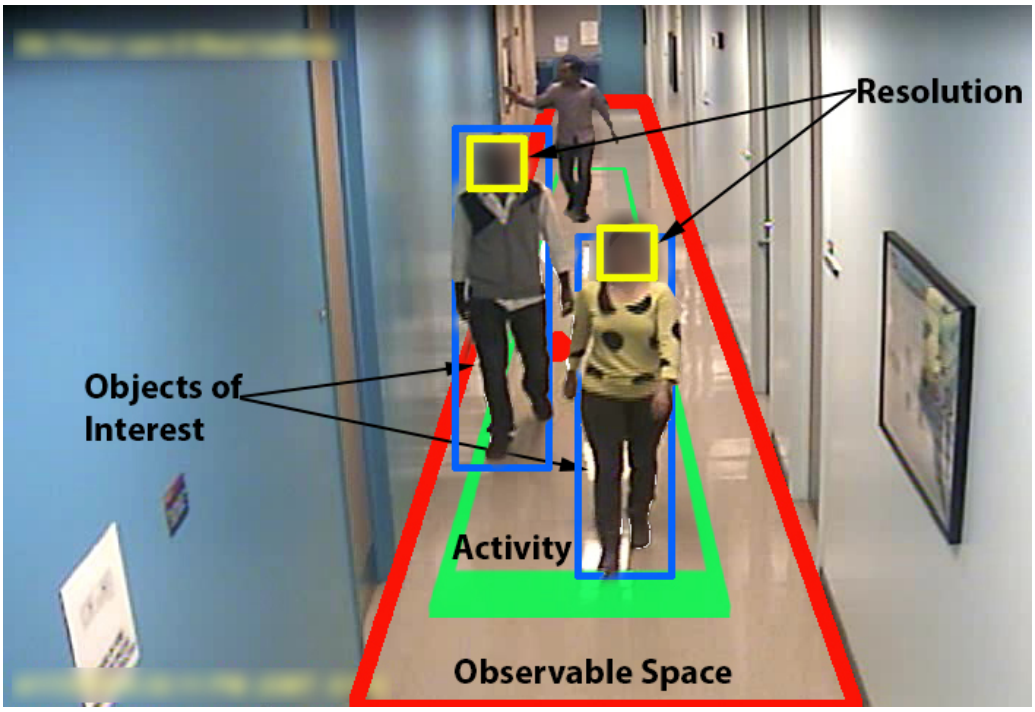


Figure 1.2: Example of an image from a single surveillance camera illustrating the four aspects of a camera view.

Consider a view from a single camera as shown in Figure 1.2. In the following, we discuss the four relevant aspects considered for an optimal camera configuration.

Maximize observable space in view: The information regarding the 3D geometry (floor) of the infrastructure can be used to maximize the observable space. In doing so, only the space that would be accessible by humans is considered relevant, as depicted by the red bounding box in Figure 1.2.

Maximize the view of regions with expected dominant human activity: Given the observable space, there are regions within it where one can expect dominant human activity to occur. This is illustrated by the green bounding box in Figure 1.2. All public infrastructures have entrances, exits and points of interest. Any doorway can be considered as an entrance or an exit. For simplicity they are referred to as nodes. In an infrastructure, different nodes are accessed with different frequencies. A node representing a common entrance or an exit has a high frequency of access as opposed to an employee's personal office. Given these nodes and their probabilities, human motion can be estimated or measured between the nodes to identify regions of high human activity.

Maximizes the ability to capture preferred pose of objects of interest: In this surveillance scenario, frontal pose of the humans can be considered to be the preferred pose as illustrated by the blue bounding boxes in Figure 1.2. The direction of motion of humans can be used to maximize the view of their frontal pose. Given the nodes, their probabilities, and trajectories followed by humans, this direction of motion can be identified.

Resolution of the imaged objects: The resolution of the face (yellow bounding box in Figure 1.2) could be considered as a feature of interest in the domain of human surveillance, and hence it's captured image resolution would be expected

to be high. Similar to the previous step, the trajectories provide the direction of motion for the humans. A location of a face can be assumed based on the estimate of the average human height. The number of rendered pixels of the bounding box representing the face in the image from the camera can be used for maximizing this quantity.

1.2.3 People Tracking

People tracking is the ability to identify the position of a specified person in the camera view with the progression of time. Tracking has applications in multiple disciplines like surveillance, robot motion planning, etc. For example, in surveillance, it can be used to monitor a scene and detect abnormal activities. In robot motion planning, it can be used to identify people and plan a path to avoid collisions [105]. Recent methods in people tracking follow a two stage cycle of detection and prediction as shown in figure 1.3. In the detection phase, an appearance model is used to describe the object of interest and the location of the object is initialized for the tracking process. In the prediction stage, a motion model is used to predict the future location of the detected object, and based on this prediction, a localized area is defined where the object might exist.

Given a good appearance model and the motion model, object initialization, localization and association can be trivial tasks. The core of the detection stage is the appearance model and that of the prediction stage is the motion model. Hence significant research has been focused on improving appearance models for detection and

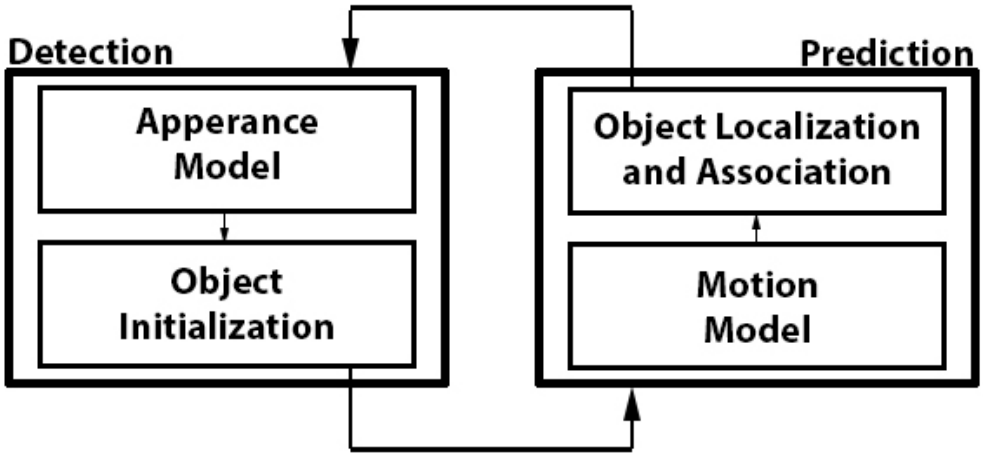


Figure 1.3: Steps involved in the tracking process.

motion models for prediction. This work proposes the use of a human motion model for prediction, which can be used in conjunction with any human appearance based detection model. Motion models assume an underlying law for predicting the future state of the object being tracked. For example, consider tracking a free falling ball. The laws of gravity can be used to generate a motion model. However, in this case the objects being tracked are humans and the design of the underlying law/system is non-trivial. Human motion can be complex, attributing to the multitude of factors that influence it, like other humans, geometry, destination, etc.

Human motion can also be effected by other social and cultural factors. In this work, we propose a motion model for prediction that accounts for geometry, objects, and other humans in the environment. The proposed method takes as input the entire 3D geometry of the environment and performs motion prediction in 3D.

1.3 Contribution

- We propose a set of novel geometric features that describe a point on the floor with respect to the geometry of the 3D environment around it considering the perception of the geometry within the context of behavioral norms.
- Given the geometry of an environment, we propose a method to estimate the human occupancy map.
- Given the geometry of an environment and the humans in it, the location of start and destination of a subject, we propose a model to forecast the trajectory of the human subject by leveraging this developed human occupancy map.
- A method to embed the forecasting along with the appearance features to enhance person re-identification.
- We propose a method to incorporate predicted human behavior for camera placement optimization.
- We propose a method to estimate areas of dominant human activity based on the 3D geometry of the infrastructure.
- We propose a method to identify and cluster regions of plausible high human activity.
- We propose a metric to assess the quality of a camera configuration based on observable space, amount of activity in the view, preferred pose of objects of interest, and their image resolution.

- We propose a method for tracking humans by leveraging the aforementioned motion estimation model which can handle occlusions and even allow for tracking across non-overlapping cameras.

Chapter 2

Background

2.1 Trajectory Forecasting

Trajectory forecasting is a widely researched field. A complete survey was done by Morris and Trivedi [70]. Traditional models for forecasting have followed a two stage approach, a data-driven learning and then a prediction stage. In the learning stage, the trajectory patterns are observed for the scenario and a model is learned. In the prediction stage, the initial information about the trajectory is used along with the learned model to predict future actions. Junejo *et al.* [52] used minimum graph cuts with edges weighted by Hausdorff distance for training and a combination of spatial, velocity and curvature features for trajectory prediction of real world outdoor pedestrians. Vasquez and Fraichard [92] used pairwise clustering to learn trajectory models of human subjects within indoor scenarios, and the prediction is done using the mean and the variance of the clusters. Weiming *et al.* [45] learnt the trajectory model of

real-world pedestrians and toy cars in a model scene using fuzzy self-organizing neural networks. Markov model was used to model the piecewise trajectories of vehicles by Paki and Martial [21]. A bank of previously observed switched dynamic models were used for predicting the trajectories of humans in indoor scenarios by Nascimento *et al.* [72]. Vasquez and Fraichard [93] used a hidden Markov model based on growing neural gas algorithm for trajectory forecasting within indoor scenarios. Saleemi *et al.* [80] modeled the trajectory patterns of real world pedestrians using a kernel density estimator and a unified Monte Carlo Markov Chain framework was used for predicting the likely trajectories. All of the above models work at a pixel level on the 2D images and does not explicitly model the effect of the environment on the humans that may influence the shaping of their trajectory. Moreover, the models are scene-dependent and cannot be transfered to a new geometry. So, even a small change in the environment like introducing a new object in the scene would require a complete new set of training data.

Recently proposed prediction models employ a model-driven approach that accounts for the environment. Bhattacharya *et al.* hypothesized the trajectories around obstacles for robot motion planning in an environment by forming homotopy classes [35, 11]. Ziebart *et al.* [112] used maximum entropy inverse optimal control for prediction of human trajectories in outdoor scenarios, and also took the environment into consideration. Kitani *et al.* [55] modeled the environment using semantic scene labeling and performed prediction using inverse optimal control. More recently in [96] a visual prediction of motion was also generated along with trajectory forecasting. While [55] is closely related to our approach, the effect of the environment

modeled on the formation of the human trajectory is limited to a small set of scene labels. The scene understanding is at an image level. In our method, we build and use the actual 3D model of the entire geometry, and the amount of training required is minimal and only done once. This is because, we are trying to learn the human behavior around the 3D geometry in general, rather than trying to learn the human behavior for a specific scene. Once we understand this behavior, the human behavior for any new geometry can be estimated without training as long as the new geometry is available. To the best of our knowledge, this is the first work that accounts for the 3D geometry for trajectory forecasting. Our method is not scene or geometry dependent and explicitly models the effect of the 3D geometry on humans to predict their likely trajectory.

2.2 Person Re-Identification

Considerable amount of research has been performed in the area of re-ID. Re-ID problems are widely viewed as recognition problems. A database of known identities is called a gallery set. Given an observation whose identity is unknown called a probe, the goal of re-ID algorithm is to rank the identities in the gallery set based on a similarity score to the probe. Re-ID approaches can be broadly categorized as appearance-based methods or context-based methods. The former only uses information regarding the appearance like color and texture to construct features that describe an identity for matching, while the latter often augments this information with context like spatial and temporal data to match with the gallery set. This work

can be categorized under the latter. A complete survey of re-ID was conducted by Gala and Shah [9].

Appearance-based methods are more commonly considered than context-based methods. This can be attributed to the unavailability of the environmental geometry and camera topography for commonly used public datasets. The proposed method suggests a technique for embedding this information to build context-based re-ID algorithms. A complete survey of appearance based methods was conducted in [82] by Satta. A huge body of work exists that employ different appearance based features like color, texture, gradient and shape for re-ID, few of which are [7, 27, 37, 84, 75, 56, 110]. Since this work employs a contextual based method, they are discussed further in detail. However, the proposed algorithm is used in conjunction with an appearance based method suggested by Bazzani *et al.* in [8], which constructs a symmetry based description to characterize the human body for re-ID.

The need to associate trajectories across multiple-camera network for tracking contributed to the genesis of contextual re-ID algorithms. These methods try to understand the relationship between the cameras in a surveillance network to estimate the space or time dependency among the observations for re-ID. Makris *et al.* in [63] suggested the need for "network calibration", that describes the association among cameras for tracking across non-overlapping cameras. The network topology is represented as a graph with nodes representing the entry/exit zones of the cameras present on the network, and the edges represented the transition time and probability between the nodes. Observed trajectories are later used as training data to learn these transition time probabilities. Javed *et al.* in [50] proposed a method

for tracking people across non-overlapping cameras by learning the inter-camera relationship through exploiting the space-time cues between them. These relations are learned in the from of probability density functions of space time parameters using kernel density estimators. In [59] the camera images are represented as time series data and then segmented into regions of similar activity. Inter-region time delay are inferred using Cross Canonical Correlation Analysis. Loy *et al.* also followed a similar approach but modeled the dependency between the regional patterns as Time Delayed Probabilistic Graphical models in [60]. Mazzon *et al.* in [67] proposed Landmark-Based model (LBM) using a rough site map, made up of the projection of the camera's field of view, the unobserved regions, marked entry/exit zones of the cameras and crossing landmarks. Human trajectories are propagated along possible paths connecting the located landmarks. Using the initial observed velocity, an estimate of the time taken for traversal is calculated and used to filter the gallery set for re-ID.

In the proposed method, a complete 3D model of the camera network environment was constructed and the cameras in the real world were calibrated and then embedded as virtual cameras in the model. This step eliminates the need for training to learn the camera network topography or relationship between the cameras. Furthermore, the trajectory forecasting model for propagating humans is based on observed human behavioral norms in contrast to LBM which employs a purely random approach.

2.3 Camera Placement Optimization

Camera placement optimization is a crucial problem in computer vision and has been explored by many researchers. Most of the early work puts emphasis on image resolution and were based on a single camera focused on a static object. The problem was to find the best position for the camera that maximizes the quality of features on an object [89, 32]. Later, Chen and Davis [19] proposed a metric based on resolution and occlusion characteristics of the object that assessed the quality of multiple camera configurations. The configuration was optimized based on this metric such that minimum occlusion would occur while ensuring a certain resolution. Mittal and Davis [69] suggested a probabilistic approach for visibility analysis that captured the observable space aspect and calculated the probability of visibility of an object from at least one camera in the configuration. Then a cost function was defined that mapped the sensor parameters to the probability and the cost function was minimized by simulated annealing.

Erdem and Sclaroff [29] suggested a binary optimization approach for the camera placement problem that captured both the observable space and resolution aspect. The polygon representing the space is fragmented into an occupancy grid and the algorithm tries to minimize the cost of a camera configuration while maintaining some specified spatial resolution. Horster and Lienhart [42, 44, 43] proposed a linear programming approach that determines the calibration for each camera in the network that maximizes the coverage of the observable space with a certain resolution.

Ram *et al.* [78] proposed a performance metric that evaluates the probability of accomplishing a task as a function of set of camera configurations. This metric took into consideration the objects of interest in the scenario and was defined to realizes only images obtained in a certain direction (frontal image of the person). Bodor *et al.* [15] proposed a method, where the goal is to maximize the aggregate observable space across multiple cameras. An objective function that quantifies the resolution of the image and the motion trajectories of the object in the scene is defined. A variant of hill climbing method was used to maximize this objective function.

Murray *et al.* [71] applied coverage optimization combined with visibility analysis to address this problem. For each camera location, the coverage was calculated using visibility analysis. Maximal covering location problem (MCLP) and backup coverage location problem (BCLP) were used to model the optimum camera combinations and locations. Malik and Bajcsy [64] suggested a method for optimizing the placement of multiple stereo cameras for 3D reconstruction. An optimization framework was defined using an error-based objective function that quantifies the stereo localization error along with resolution constraints. A genetic algorithm was used to generate a preliminary solution and later refined using gradient descent. Kim and Murray [53] also employed BCLP to solve the camera coverage problem. They suggested an enhanced representation of the coverage area by representing it as a continuous variable in contrast to a commonly used discrete variable. Yabuta and Kitazawa [102] and Debaque *et al.* [25] also employed a combination of MCLP and BCLP for solving the optimum camera coverage problem. The former took into consideration the 3D

geometry of the environment and supplemented the MCLP/BCLP problem by including a minimal localization error variable for both monoscopic and stereoscopic cameras. The optimization problem was solved using simulated annealing. In the latter, the MCLP/BCLP problem was supplemented using visibility analysis for optimization. Huang *et al.* [46] proposed a two-stage approximation algorithm, the first part proposes a solution for the minimum watchmen tour problem and placed cameras along the estimated tour, the second part finds the solution to art gallery problem and added extra cameras to connect the guards. Most of the previous work emphasizes the importance of maximizing observable space and resolution of this space. There is little work addressing the significance of activity in the observable space along with obtaining useful data. This work address this by assuming equal importance to all four aspects which were ignored in the previous work.

Considering the 3D geometry of the environment is of significant value for the camera coverage optimization problem. In this work, we focus on indoor scenarios and assume the availability of a complete 3D model of the environment where the camera network is to be deployed. To the best of our knowledge, this is the first work that takes into consideration the human activity in the scenario for designing an optimal camera network in the absence of any observations. Although [15, 49] proposed the use of observed human activity for optimizing the camera placement, in the proposed work the human trajectories are simulated and not observed in order to identify regions with dominant human activity. Furthermore, Ram *et al.* [78] proposed the use of frontal view from observations as a task for optimizing the camera position unlike the proposed method that predicts frontal view based on

human behavior. Finally, the human behavior in a given scenario is influenced by the 3D geometry of that environment [65, 54]. To the best of our knowledge, this is the first work that incorporates this information to optimize the camera network locations for video surveillance.

2.4 People Tracking

Tracking is an important low level algorithm for numerous applications. Accounting to this, an immense value of research has been conducted in this area. Yilmaz *et al.* [105] and Watada *et al.* [98] performed a broad survey in object and human tracking respectively. As mentioned earlier, appearance models and motion models are the core of the detection and prediction stages. As work is focused on the prediction stage for tracking, detection methods are first briefly discussed followed by a detailed survey on existing prediction methods for tracking.

2.4.1 Detection methods

The objective of the detection phase is to identify the location of the objects of interest in the image. Yilmaz **et al.** [105] categorized the detection methods as follows:

2.4.1.1 Points based methods

The objects to be tracked are represented by a set of one or more interest points. Image features like contours of lines and end points [107, 97, 18] or color and contrast of object intensities [106, 34] were used to identify points of interest.

2.4.1.2 Segmentation based methods

The objective of segmentation is to partition the object to be tracked from the image. Commonly used segmentation techniques for tracking were the mean shift algorithm [22, 16] and histograms [113, 94, 85].

2.4.1.3 Supervised learning based methods

These methods use a dataset representing the object to train a classifier for identifying the object of interest. This classifier was used to detect the regions with object of interest in the images for tracking [73, 76].

2.4.1.4 Background subtraction methods

The objective is to isolate the foreground pixels or the objects of interest by identify and removing all the background pixels. This was the popular and conventional technique for tracking [88, 81, 108].

2.4.2 Prediction methods

Majority of the early methods used for tracking were based on detection alone, however, methods proposed latter supplemented the detection methods with the prediction phase for estimating the future location of the object. This allowed for faster tracking methods for two reasons, first, the prediction provided with an approximate location of the object for the detection algorithm and second, detection algorithm had to be run on every few frames as opposed to every frame since the location of the object could be predicted. Motion models can be designed at a 2D level on the image plane or at a 3D level on the ground plane. In general, tracking in 3D can have an advantage over 2D when handling occlusions. The proposed model performs tracking in 3D on the ground plane and hence these methods are discussed more in detail than model performing tracking in 2D.

2.4.2.1 Prediction on image plane

In these methods, the first few frames were used to learn the motion of the object and then a statistical algorithm was initialized based on the learned motion to predict the future states of the object. The most commonly applied techniques were Kalman filter [99, 47, 68] and particle filter [48, 101, 104]. Although these methods can handle occlusion to a certain extent better than detection methods, they perform poorly when tracking an object with complex motion like humans. For further reading on this methods or detections method, the readers can refer to [98, 105].

2.4.2.2 Prediction on the ground plane

A fair component of work in this area has been conducted in the robotics community, as laser range sensors allowed for a natural way to work in 3D on the ground plane in contrast to a video sensor in computer vision which required calibration and homography mapping. These methods can further be sub-categorized as follows.

Non-behavioral models: These models do not account for the complexity of human behavior and assume a linear interpolation or constant velocity for prediction. Fod *et al.* [33] proposed the use of a constant velocity model for prediction in a laser range sensor environment. The previous scans were used to estimate the velocity of the object and a Kalman filter was used to estimate the future position for tracking. Schulz *et al.* [83] introduced sample-based Joint Probabilistic Data Association Filters (SJPDAFs) for people tracking using a laser sensor which uses particle filter to track the state of the object and apply (JPDAFs) for association. Similar to Kalman filters, the previous measurements were used for prediction using a particle filter. Cui *et al.* [23] demonstrated tracking using rows of laser scanners and a video camera by employing a common Kalman filter for prediction. Arras *et al.* [4] also assumed a constant velocity model to track people's legs with laser scanner using Kalman filter and also explicitly handled occlusion for tracking. These models can cope with occlusion to a certain extent and can be used for tracking objects with linear motion. However they might not be sufficient to track humans as they exhibit complex motion.

Human behavioral models: These models either learn human behavior from observation or explicitly model it for prediction. In the former, human trajectories are observed in the scenario and a motion pattern is learned for prediction in contrast to the latter where the influence of the various factors (geometry, objects and humans) on human motion are explicitly modeled.

Liao *et al.* [57] used the floor map of the environment to generate a Voronoi graph and assumed that people travel along the edges of the map. Observed motion patterns with a laser scanner on a robot was used to calculate the transition probabilities along the edges of the Voronoi graph. These probabilities along the graph were used for prediction in tracking. Bruce and Gordon [17] observed people trajectories using laser sensor on a robot to learn destinations in the environment. Later the human motion is predicted to an estimated goal location along the path predicted by a planner. Bennewitz *et al.* [10] clustered observed human trajectories from a laser range sensor and employed Expectation-Maximization algorithm to form motion patterns. Hidden Markov model was used to predict the future states of people for tracking. Weser *et al.* [100] proposed the use of self organizing maps to learn motion patterns from trajectories obtained from a laser range sensor. A particle filter was used to predict the future position of humans using the learned motion patterns for tracking. These models can handle occlusions in static environments but fail when deployed in a dynamic environment with moving humans because the learned motion pattern is not accurate anymore. Furthermore introduction of a new static object in the environment would require new observations for training and generating the motion patterns.

These drawbacks were overcome by explicitly modeling human behavior and supplementing it with observed motion patterns for prediction. Antonini *et al.* [2] proposed the use of discrete choice models with varying velocity options to build a probability distribution and sample the future state by accounting for other humans and environment. The predictions were used for detection and tracking in a video sequences. Pellegrini *et al.* [74] proposed Linear Trajectory Avoidance (LTA) model taking into consideration other humans and static objects such that the pedestrians steer clear to avoid collision. The model was incorporated for tracking in video data. Yamaguchi *et al.* [103] defined an energy function that evaluates the future states based on destination, other humans, static objects and group behavior. An energy minimization framework was used for predicting the future states. The results were demonstrated using a tracking algorithm in video sequences. Luber *et al.* [62] proposed the use of Social Force Model [41] for tracking in data collected by laser scanner and video data. Social Force model was one of the earliest work in human motion dynamics which modeled the interaction between humans, objects and geometry as repulsion and attraction forces. Gong *et al.* [36] implemented Multi-Hypothesis motion planning for video tracking. This model takes into account the geometry and hypothesizes multiple routes around objects, but fails to model the social interaction between objects and humans. Luber *et al.* [61] generated a spatial affordance map which represented the global human activity of the environment assuming events occur as a Poisson's process. This map was incorporated into a multi-hypothesis tracker for enhancing motion prediction for tracking using a laser range sensor.

CTF generates an occupancy map for any geometry based on observed human

behavior. This model takes into account the static objects and geometry and predicts trajectory to a given destination. In the proposed method the model is enhanced to handle social interaction with humans and incorporated into a tracking algorithm.

Chapter 3

Contextual Trajectory Forecasting

3.1 Problem Formulation

Given the 3D geometry of the environment like the floors, walls, hallways, etc. along with the starting point and the destination of a trajectory. CTF is modeled as a Markov chain model. Let $P = \{p_1, p_2, p_3...\}$ be set of all points on the floor like the centroids of triangles in a triangle mesh as shown in Figure 3.1. The motion from a starting point to a destination point is depicted as a trajectory T formed by transitions from one point to another. $T = \{S_1, S_2, S_3...\}$ where S_1, S_2, S_3 are the states at times $\{t_1, t_2, t_3...\}$, and $S_i \in P$. As in a Markov chain model, the decision of which point to transition next depends only on the current state of the subject and can be denoted by:

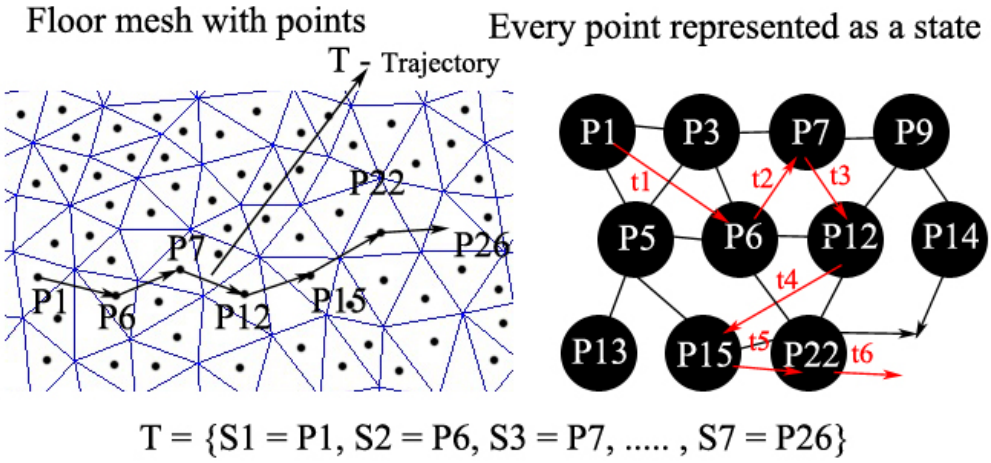


Figure 3.1: Trajectory modeled as a Markov chain model.

$$P(S_n|S_{n-1}, S_{n-1}|S_{n-2}...) = P(S_n|S_{n-1})P(S_{n-1}|S_{n-2})... \tag{3.1}$$

The Scenario in Figure 3.1 can be model as a Morkov Chain model:

$$\begin{aligned} P(S_7 = P_{26}|S_6 = P_{22}, ..., S_1 = P_1) &= P(S_7 = P_{26}|S_6 = P_{22})...P(S_2 = P_6|S_1 = P_1) \\ &= P(S_7 = P_{26}|S_6 = P_{22}) \end{aligned}$$

The problem is to create a transition matrix $P(S_n|S_{n-1})$ that the Markov model can use to sample points to form trajectories that travel from the start to destination and also conform with behavioral norms.

The steps involved in CTF model are (1) estimate the occupancy map of the new geometry (2) create the distance map based on the destination (3) combine the occupancy map and the distance map to create the energy function (4) define a transition matrix based on an energy maximization framework (5) sample points

using the transition matrix to form a trajectory (6) and then use the predicted trajectories. Given the geometry, the occupancy map is estimated first. Later given the destination, the distance map, the energy function and consequently the transition matrix are estimated. The method is described in the following sections as (a) Section 3.2 describes estimating the occupancy map and (b) Section 3.3 describes trajectory forecasting. The flowchart in Figure 3.1 showcases the entire framework.

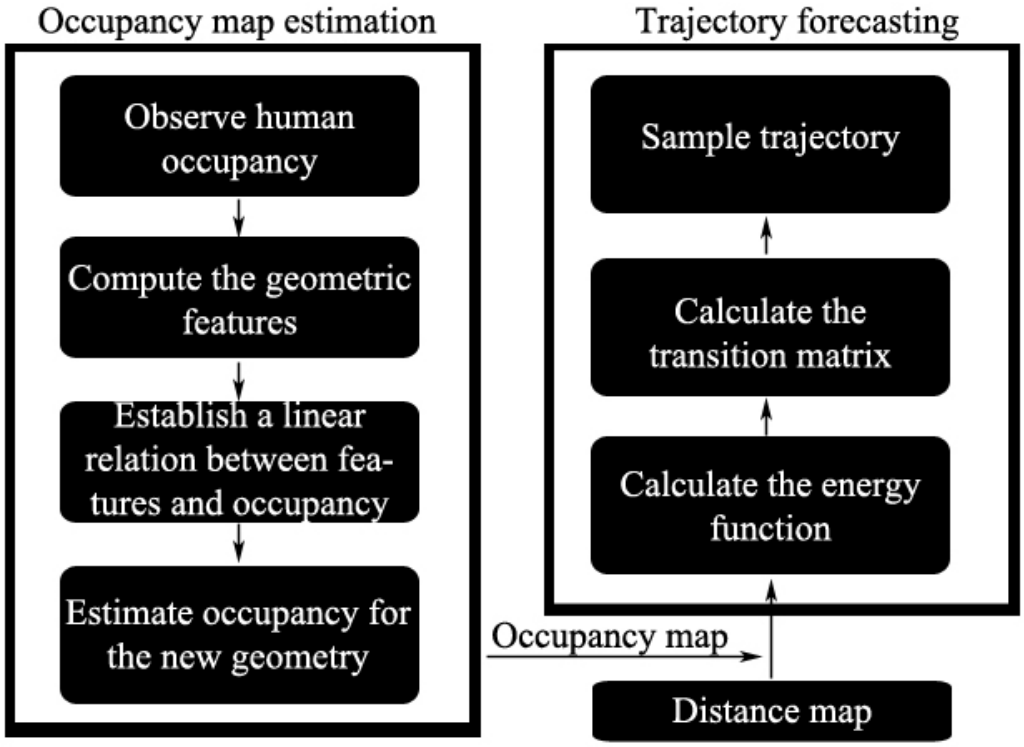


Figure 3.2: Flowchart illustrating the trajectory forecasting framework.

3.2 Occupancy Map Estimation

To estimate the occupancy map for a novel geometry, we begin by observing the occupancy map for a known geometry. Then a certain set of geometric features are computed for the points on the floor with respect to the surrounding geometry. Later a relationship is established between the occupancy of the point and its computed features. This relationship is leveraged to estimate the occupancy map for any new geometry.

3.2.1 Observing the Human Occupancy Map

In the geometry, the floor was modeled as a uniform triangle mesh. Let the centroids of the triangles on the floor mesh be represented by a set of points P. The video from a calibrated camera was obtained for a prolonged period of time, and then human detection [24] was performed, that outputs the bounding box for each detected human. For every detected human, the bottom of the bounding box was re-projected onto the floor in the 3D model. The occupancy of the triangle in which this re-projected point falls was increased accordingly. The occupancy map observed in a hallway over a period of 5 days is shown in Figure 3.3.

3.2.2 Geometric Features

The features f_i of any point p_i on the floor in the 3D model are represented as a set of numbers $\{d_{i1}, d_{i2}, d_{i3}...\}$, which are its distances from the walls and objects

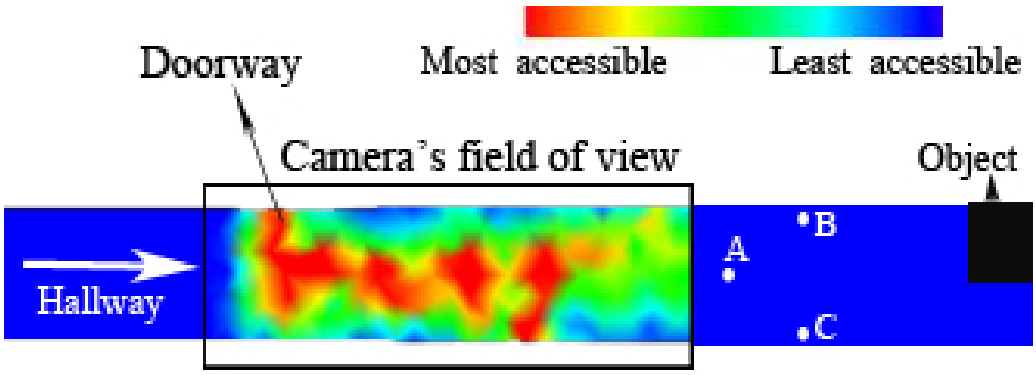


Figure 3.3: Observed occupancy map of a hallway in a building from a video observed over 5 days.

surrounding the point p_i. The richest description is obtained by taking distances from the point on the floor to every other point in the geometry surrounding it. However, such amounts of data is redundant, computationally infeasible and will likely result in over training. The feature set should contain sufficient information to estimate the occupancy of that point. When traversing indoors, a humans immediate decision of movement is influenced by the objects in their path in the hallway and the surrounding walls. For example the way humans navigate around tables and chairs when moving from one corner of a classroom to the opposite corner. So, to build these features, distance was measured to walls or objects in the hallway along vectors pointing at a certain inclination from the floor. In this work, a 30 - 60 degrees inclination was used, considering this was sufficient to capture objects present in the hallway. Pointing vectors at regular interval spanning an entire circle with its tail fixed at the point p_i as shown in Figure 3.4. There are two issues concerning these features. First, if the closest wall or an object in a certain direction is very far away

like in the case of an object at the far end of a hallway like the point A in Figure 3.3 with respect to the object, the local motion or occupancy decisions of a human is indifferent to an object at such great distance. Second, consider two points on the floor that are close to the walls in a hallway but on either end of the hallway like the points B and C in Figure 3.3, these points in essence are the same and are likely to have the same occupancy, yet the features representing these points are different. For example when measuring distances surrounding the point in clockwise direction starting from the first direction pointing upwards, the features of B would start with a small number and increase before decreasing. However for C, the features would start with a much larger value and then increase before decreasing to a smaller value following the convention for computing distances. So the features would require some preprocessing as we would like to make the features scale and rotational invariant.

Scale invariance can be achieved by thresholding the distances to a hemisphere with its center at the location of the human subject's feet as shown in Figure 3.4. The radius r of this hemisphere was inferred from the theory of Proxemics [38]. This is a theory based on observation that defines how human beings unintentionally make use of physical space around them. Proxemics classifies the space close to a human subject into four broad regions, Intimate, Personal, Social and Public distance. The interaction between human subjects in closed hallways was assumed to take place within the social distance, which is 7-12 ft. (80-140 in.). The radius of the hemisphere was defined by this distance ($f_i = \{d_{i1}, d_{i2}, d_{i3}...\}, d_{ij} = r \forall d_{ij} \geq r, 80 \leq r \leq 140$). To make the features rotationally invariant, the distances were always measured starting with the smallest distance ($f_i = \{d_{i1}, d_{i2}, d_{i3}...\}, d_{i1} \leq d_{ij}, 2 \leq j \leq n$)

and furthermore the measurements were taken following either a clockwise or anti-clockwise convention, in which case, the points B and C in Figure 3.3 will have similar features. The features were not arranged in ascending order, but were only measured starting from the smallest value keeping the order unchanged.

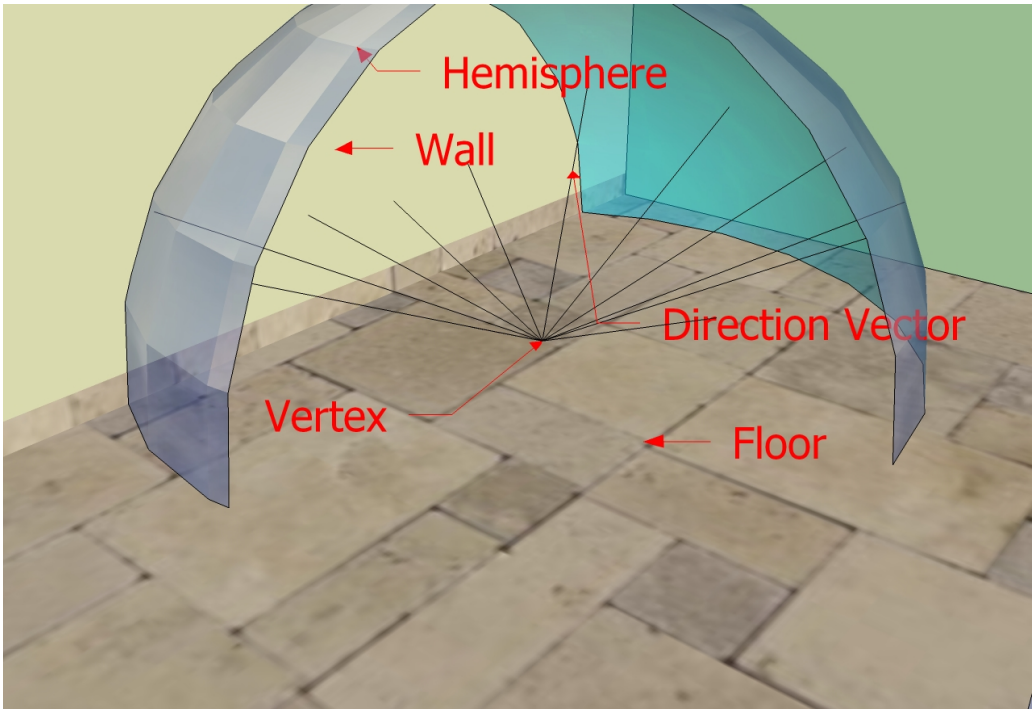


Figure 3.4: Geometric features.

3.2.3 Modeling Relationship between Occupancy Map and Geometric Features

Now that the geometry and its humans occupancy behavior are available, a relationship was modeled between them using linear regression and support vector machine regression. To build a linear relationship, let $f_i = \{d_{i1}, d_{i2}, d_{i3}...\}$ be the features of

the points p_i with occupancy o_i. Given the dataset $\{o_i, d_{i1}, d_{i2}, ...d_{in}\}$, where o_i is the dependent variable and the vectors f_i are the independent variables. If e_i is the error term, the relationship can be expressed as a set of linear equations.

$$o_i = \beta_1 d_{i1} + \beta_2 d_{i2} + ... + \beta_n d_{in} + e_i = F_i^T \beta + e_i \tag{3.2}$$

$$o = F\beta + E \tag{3.3}$$

where,

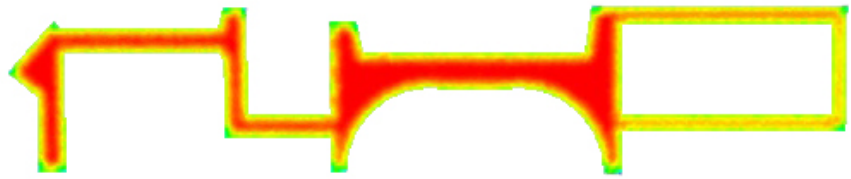
$$o = \begin{pmatrix} o_1 \\ o_2 \\ . \\ . \\ . \end{pmatrix}; \quad F = \begin{pmatrix} f_1 \\ f_2 \\ . \\ . \\ . \end{pmatrix} = \begin{pmatrix} d_{11} & ... & d_{1n} \\ d_{21} & ... & d_{2n} \\ . & ... & . \\ . & ... & . \\ . & ... & . \end{pmatrix};$$

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ . \\ . \\ . \end{pmatrix}; \qquad \mathrm{E} = \begin{pmatrix} e_1 \\ e_2 \\ . \\ . \\ . \end{pmatrix}$$

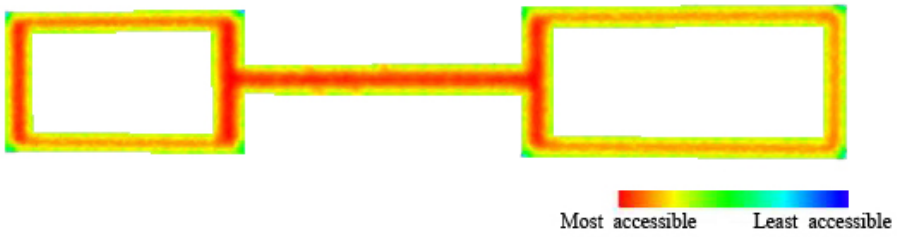
Minimizing the sum of squares of the error term E to estimate β,

$$\beta = (F^T F)^{-1} F^T o \tag{3.4}$$

To estimate the occupancy of a point on the floor in a new geometry, first the geometric features were computed and then the estimated β values were substituted in Equation 3.2. Figure 3.5 depicts the occupancy for two different geometries in a building.



(a)



(b)

Figure 3.5: Estimated occupancy maps through linear regression using 12 features and radius 60: Red being most accessible and blue being the least. (a) geometry A; (b) geometry B.

The occupancy map was also estimated using support vector machine regression as described by Smola and Scholkoph in [87]. The observed occupancy and the calculated features was used to create a training dataset defined by $\{(f_1, o_1), (f_2, o_2), ..., (f_l, o_l)\}$. In ϵ-SV regression, the goal is to estimate a function $h(f)$ such that it has at the most a deviation of ϵ from the observed occupancy o_i for all the training data, and also is as flat as possible. Considering a linear function

$$h(f) = \mathbf{w} \cdot \mathbf{f} + b, b \in \mathbb{R} \tag{3.5}$$

Vapnik in [91] described this as an optimization problem and also introduced slack

variables ξ_i, ξ_i^* to cope with infeasible constraints. The formulation is stated as:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ \\ & o_i - \mathbf{w}\cdot\mathbf{f} - b \leq \epsilon + \xi_i \\ \text{subject to} \quad & \mathbf{w}\cdot\mathbf{f} + b - o_i \leq \epsilon + \xi_i^* \\ & \xi_1, \xi_i^* \geq 0 \end{aligned}$$

This is referred to as the primal problem. The constant $C > 0$ determines the trade off between the flatness of h and the amount up to which deviations larger than ϵ are tolerated. This problem is solved by rewriting it in its dual form. A Lagrangian function is constructed using a dual set of variables.

$$\begin{aligned} L := \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ - \sum_{i=1}^{l}\alpha_i(\epsilon + \xi_i - o_i + \mathbf{w}\cdot\mathbf{f_i} + b) \\ - \sum_{i=1}^{l}\alpha_i^*(\epsilon + \xi_i^* + o_i - \mathbf{w}\cdot\mathbf{f_i} - b) \end{aligned} \tag{3.6}$$

For optimality, the partial derivatives of L should vanish with respect to the primal variables (w, b, ξ_i, ξ_i^*).

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{3.7}$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)f_i = 0 \tag{3.8}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \eta_i = 0 \tag{3.9}$$

$$\frac{\partial L}{\partial \xi_i^*} = C - \alpha_i^* - \eta_i^* = 0 \tag{3.10}$$

It is shown that the dual variable α_i^*, α_i can be obtained by solving the dual problem.

$$\text{maximize} \quad \begin{array}{l} -\frac{(1)}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) f_i \cdot f_j \\ -\epsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i,j=1}^{l} o_i(\alpha_i - \alpha_i^*) \end{array}$$

using equation Equation 3.8,

$$\text{subject to} \quad \begin{array}{rcl} \sum_{i,j=1}^{l} o_i(\alpha_i - \alpha_i^*) & = & 0 \\ \alpha_i, \alpha_i^* & \in & [0, C] \end{array}$$

we can write Equation 3.5 as

$$h(f) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) f_i \cdot f + b \tag{3.11}$$

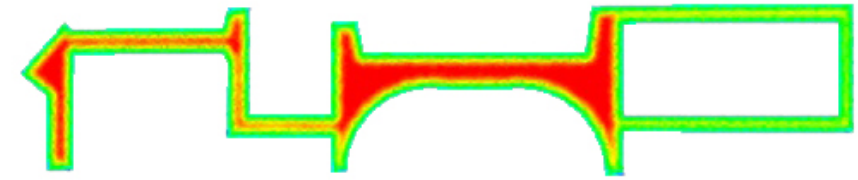
Finally it is shown that b can be computed by exploiting the Karush-Kuhn-Tucker conditions. It is shown that

$$\begin{array}{c} max\{-\epsilon + o_i - w \cdot f_i | \alpha_i < C, \alpha_i^* > 0\} \leq b \leq \\ min\{-\epsilon + o_i - w \cdot f_i | \alpha_i > 0, \alpha_i^* > C\} \end{array} \tag{3.12}$$

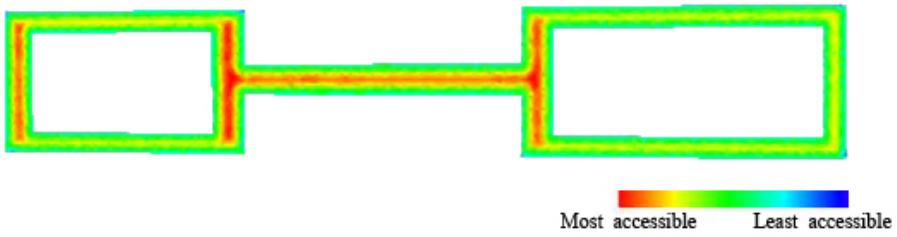
Once b is chosen based on the above conditions, the value of occupancy for any point on the floor whose geometric features are known can be calculated using Equations 3.11. Figure 3.6 depicts the occupancy for two different geometries in a building.

3.3 Trajectory Forecasting

Once the occupancy map for a new geometry was estimated, the next step was to calculate the distance map based on the given destination. Then finally a energy maximization framework was used to create the transition matrix for trajectory forecasting.



(a)



(b)

Figure 3.6: Estimated occupancy maps through support vector regression using 12 features and radius 60: Red being most accessible and blue being the least. (a) geometry A; (b) geometry B.

3.3.1 Destination Map

A distance map was created by calculating the distance to the destination for every point on the floor. In general, hallways are complex polygons with areas that are inaccessible. Using Euclidean distance can potentially be erroneous. Geodesic distance as defined in [66] was used instead. Euclidean distance between two points is not altered by the presence of inaccessible areas, but geodesic distance is measured around the inaccessible areas along the hallway and gives a more accurate sense of distance for human navigation. A rendering of the distance map for geometry A with

a given destination is shown in Figure 3.7 (a).

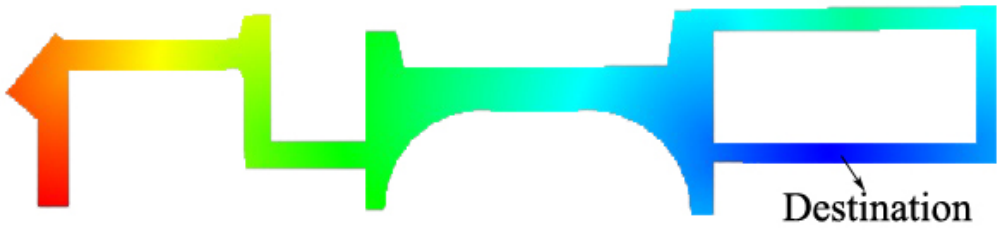


Figure 3.7: Distance map for geometry A with a given destination: Red represents the farthest points and blue the closest.

3.3.2 The Energy Function

A combination of the distance map and the occupancy map was used to create an energy function. Let O be the occupancy map function and let D be the distance map function. If $p_i \in P$ is any point on the floor. Then the energy of that point is defined by the function $E = -D(p_i)/O(p_i)$. The energy function for geometry A is shown in Figure 3.8 (b). The obtained energy function assigns higher values to points in the center of the hallway than along the edges, and the energy keeps increasing towards the destination.

3.3.3 Trajectory Sampling

The transition probability matrix was built by choosing states that maximize the energy with higher probability. For every state the subject is present in, the only

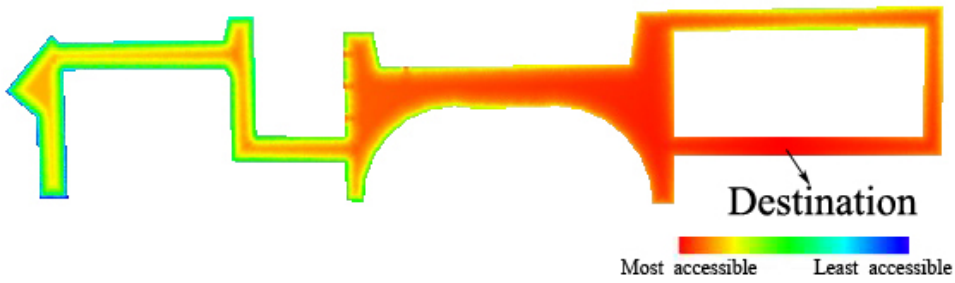


Figure 3.8: Energy function for geometry A with a given destination.

possible states of transition are states representing its neighboring points. Let the current state be S_t, and this point has m neighbors $S_{t1}, S_{t2}, ...S_{tm}$. The probability of transition to these m neighboring states is proportional to the difference in energy. So $P(S_{tm}|S_t)$ is.

$$\propto \left\{ \begin{array}{cc} E(s_{tm}) - E(s_t) & \text{if} \quad D(s_{tm}) - D(s_t) \leq 0 \\ 0 & otherwise \end{array} \right\} \tag{3.13}$$

Only states that are closer to the destination were allowed to be chosen (i.e. $D(s_{tm}) - D(s_t) \leq 0$), to ensure that the propagation does not get stuck in a local maximum. The neighboring states are sampled with a probability that is proportional to the difference in their energies.

For example in figure 3.9 the neighbors of the point p_6 are shaded. These are the only possible states of transition. The probability of transitioning from p_6 to p_7, $P(S_{p_7}|S_{p_6})$ would be proportional to:

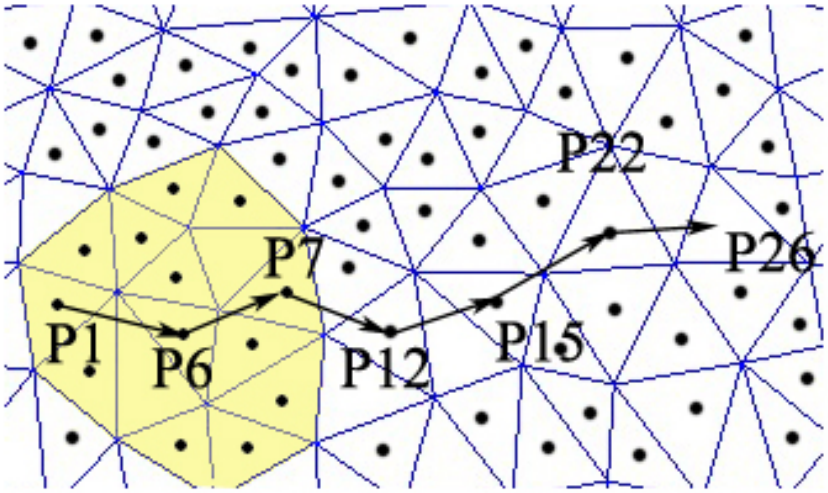


Figure 3.9: Sampling Neighbors for Transition

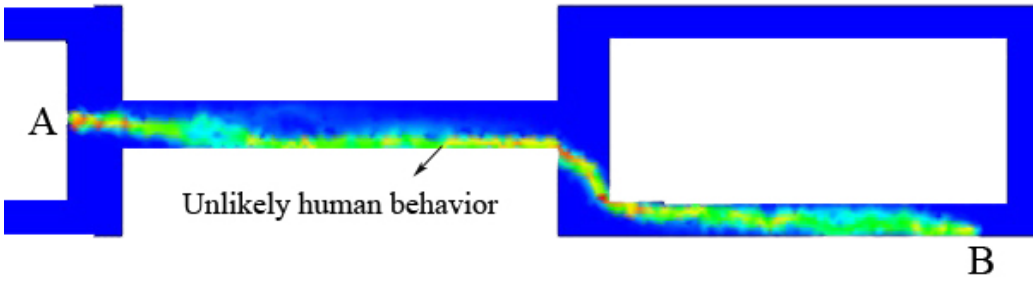
$$\propto \left\{ \begin{array}{cc} E(s_{p_7}) - E(s_{p_6}) & \text{if} \quad D(s_{p_7}) - D(s_{p_6}) \leq 0 \\ 0 & otherwise \end{array} \right\} \tag{3.14}$$

Algorithm 1 summarizes our complete trajectory forecasting method.

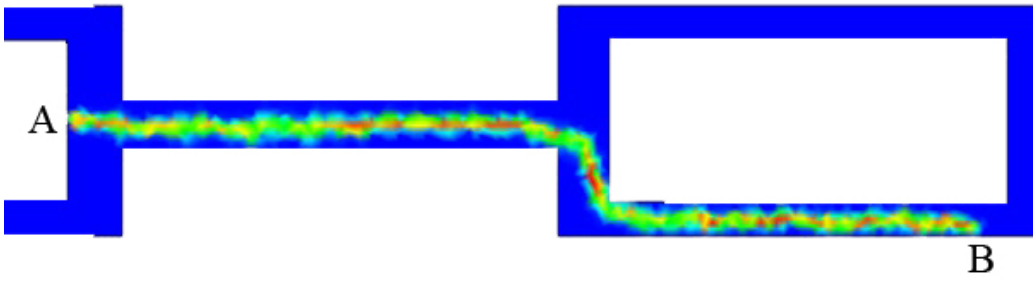
Algorithm 1 Trajectory Forecasting Algorithm

procedure TRAJECTORY–SAMPLING
    T be the Trajectory;
    S_s be the starting state;
    S_d be the destination state;
    S_t be any temporary state;
    Add S_s to T;
    $S_t = S_s$;
    while $S_t \neq S_d$ **do**
        for all neighbors $\{S_{t1}, S_{t2}...\}$ of S_t **do**
            if $D(S_t) \geq D(S_{ti})$ **then**
                $P(S_t|S_{ti}) \propto E(S_{ti}) - E(S_t)$;
            else
                $P(S_t|S_{ti}) = 0$;
            end if
        end for
        Sample the neighbor S_{ti} with probability $P(S_t|S_{ti})$;
        Add S_{ti} to T;
        $S_t = S_{ti}$;
    end while
end procedure

Figure 3.10 (a) shows the distribution of simulating the trajectory prediction algorithm 5,000 times without the use of the occupancy map, but only using distance minimization. Figure 3.10 (b) simulates with the help of the occupancy map in geometry B. It showcases how the estimated occupancy map complements the geodesic distance minimization and forms a more desirable trajectory, that conforms to expected human behavior.



(a)



(b)

Figure 3.10: A is the starting location and B is the destination (a) Distribution created by simulating trajectory prediction without using occupancy map; (b) Distribution created by simulating trajectory prediction using occupancy map.

Chapter 4

Applications

4.1 Person Re-Identification

Let $\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_m\}$ be the gallery set of m known identities, and $\Phi = \{\phi_1, \phi_2, ..., \phi_n\}$ the set of n probes. For every probe $\phi_i \in \Phi$, the problem is to rank the gallery set as $\{\gamma_{i1}, \gamma_{i2}, ..., \gamma_{im}\}, \gamma_{ij} \in \Gamma$ based on their matching score to the probe ϕ_i. Let $\zeta_x = \{c_x, a_x\}$ be the features of the identity $x \in \{\Gamma \cup \Phi\}$, where c_x are the contextual feature and a_x are the appearance features. Let $c_x = \{l_x, v_x, t_x\}$ be the contextual features of x observed at location l_x traveling with velocity v_x at time t_x. This work describes a method of leveraging contextual features to be used in conjunction with existing appearance based method and hence a_x is described by the chosen method. The matching function $M_{ij} = M(\phi_i, \gamma_j) = M(\zeta_{\phi_i}, \zeta_{\gamma_j}) = M\left(s_c(c_{\phi_i}, c_{\gamma_j}), s_a(a_{\phi_i}, a_{\gamma_j})\right)$ calculates the matching score of the probe ϕ_i to gallery item γ_j, where s_c and s_a are scores estimated on the contextual and appearance features respectively. The gallery

items for the probe ϕ_i are ranked as $\{\gamma_{i_1}, \gamma_{i_2}, ..., \gamma_{i_m}\}$ such that $M_{i_1 j} < M_{i_2 j} < ... < M_{i_m j}$.

To estimate the score s_c, consider a probe ϕ with features $c_\phi = \{l_\phi, v_\phi, t_\phi\}$ to be compared with gallery item γ with feature $c_\gamma = \{l_\gamma, v_\gamma, t_\gamma\}$. let $T_{\gamma\phi} = \{({l_1}^{\gamma\phi}, {t_1}^{\gamma\phi}), ({l_2}^{\gamma\phi}, {t_2}^{\gamma\phi}), ..., ({l_r}^{\gamma\phi}, {t_r}^{\gamma\phi})\} | ({l_1}^{\gamma\phi}, {t_1}^{\gamma\phi}) = (l_\gamma, t_\gamma), ({l_r}^{\gamma\phi}, {t_r}^{\gamma\phi}) = (l_\phi, t_\phi)$ be the trajectory that the subject has taken to reach the location l_ϕ at time t_ϕ starting from the location l_γ at time t_γ. If the trajectory was known, the probe ϕ can be associated with the correct gallery item $\gamma^{'}$ by following it. It is not possible to observe the trajectory across non-overlapping cameras. Given the geometry, the idea is to predict the trajectory using CTF from the gallery set to the probe, to find the best match in space and time. CTF provides a prediction for $T^{'}_{\gamma\phi}$ from which the contextual score $s_c(\phi)$ can be calculated.

Let the points predicted by CTF from l_γ to l_ϕ be $\{l_\gamma, l_2, ..., l_\phi\}$. Assuming that the human subject moves at a constant velocity v_γ, the time t_i taken to reach location l_i from l_γ can be estimated as $t_i = t_\gamma + \frac{d(l_i, l_\gamma)}{v_\gamma}$, where $d(l_i, l_\gamma)$ is the length of the trajectory from l_γ to l_i. CTF predicts an estimate of the trajectory from gallery γ to probe ϕ as $T^{'}_{\gamma\phi} = \{(l_\gamma, t_\gamma), (l^{'}_2, t^{'}_2), ..., (l_\phi, t^{'}_r)\}$. The contextual score of the probe ϕ and the gallery γ are defined as

$$s_c(\phi) = t_\phi; s_c(\gamma) = t^{'}_r \tag{4.1}$$

Symmetry-Driven Accumulation of Local Features (SDALF) is a symmetry based description of the human body. In SDALF, the asymmetry principles allows the segregation of meaningful body parts (head, upper body and lower body). The

symmetry criteria helps in extracting the actual appearance features. SDALF uses three different appearance features. First a HSV histogram is used to capture the global chromatic content, second, Maximally Stable Color Regions (MSCR) is used to capture the pre-region color displacement and finally Recurrent Highly Structured Patches (RHSP) are estimated by a per-patch similarity analysis. Let $s_a = \{s_a^{WHSV}, s_a^{MSCR}, s_a^{RHSP}\}$ be the appearance score values. If $\{d_{WHSV}, d_{MSCR}, d_{RHSP}\}$ be the distance functions that calculate the HSV, MSCR and RHSP distance between the probe and gallery items, then SDALF matching distance is defined as convex combination of these features.

$$\begin{aligned} d(\phi,\gamma) = \rho_{WHSV} \cdot d_{WHSV}(s_a^{WHSV}(\phi), s_a^{WHSV}(\gamma)) + \\ \rho_{MSCR} \cdot d_{MSCR}(s_a^{MSCR}(\phi), s_a^{MSCR}(\gamma)) + \\ \rho_{RHSP} \cdot d_{RHSP}(s_a^{RHSP}(\phi), s_a^{RHSP}(\gamma)) \end{aligned} \tag{4.2}$$

Where ρ are the weighting parameters. The contextual distance function is defined as $d_{CTF}(\phi,\gamma) = d_{CTF}(s_c(\phi), s_c(\gamma)) = |t_\phi - t_r'|$, t_r' and t_ϕ are as defined in Equation 4.1. The CTF distances were normalized such that $d_{CTF} \in \{0,1\}$. The CTF score is embedded in Equation 4.2 as:

$$\begin{aligned} d(\phi,\gamma) = \rho_{WHSV} \cdot d_{WHSV}(s_a^{WHSV}(\phi), s_a^{WHSV}(\gamma)) + \\ \rho_{MSCR} \cdot d_{MSCR}(s_a^{MSCR}(\phi), s_a^{MSCR}(\gamma)) + \\ \rho_{RHSP} \cdot d_{RHSP}(s_a^{RHSP}(\phi), s_a^{RHSP}(\gamma)) + \\ \rho_{CTF} \cdot d_{CTF}(s_c(\phi), s_c(\gamma)) \end{aligned} \tag{4.3}$$

In the experiments, we fix the values of the parameters as follows: $\rho_{WHSV} = 0.03, \rho_{MSCR} = 0.03, \rho_{RHSP} = 0.03, \rho_{CTF} = 0.9$. These values seems to provide the

best performance. The high value of ρ_{CTF} compared to other parameters allows for temporally constraining the data and then trying to find the best match using SDALF within the temporally constrained data. The matching function ranks the gallery items for probe ϕ as $\{\gamma_{\phi_1}, \gamma_{\phi_2}, ..., \gamma_{\phi_m}\}$ such that $M_{\gamma_1\phi} < M_{\gamma_2\phi} < ... < M_{\gamma_m\phi} \equiv d(\phi, \gamma_1) < d(\phi, \gamma_2) < ... < d(\phi, \gamma_m)$.

4.2 Camera Placement Optimization

4.2.1 Problem Formulation

Let G be the geometry (floors, ceilings, walls, etc.) of an infrastructure. Let $\{C_1, C_2, ..., C_\nu\}$ be the a set of cameras located in G with configurations (like position, orientation, zoom, etc.) represented by $\{\omega_1, \omega_2, ..., \omega_\nu\}, \omega_i \in \Omega$, where Ω is the set of all possible configurations within G. Let $g : \omega \mapsto \mathbb{R}$ be an objective function. The problem is to find a set of optimal configurations $\{\omega_1^*, \omega_2^*, ..., \omega_\nu^*\}$ such that:

$$\{\omega_1^*, \omega_2^*, ..., \omega_\nu^*\} = \underset{\{\omega_1, \omega_2, ..., \omega_\nu\} \in \Omega}{\arg\max} \sum_{i=1}^{\nu} g(\omega_i) \tag{4.4}$$

4.2.2 Camera Coverage Quality Metric

The function $g(.)$ quantifies the following aspects in view of the camera

- amount of observable space,

- amount of view of regions with expected dominant activity,
- amount of ability to capture the preferred pose of objects and
- image resolution of these objects.

Janoos *et al.* [49] proposed cell coverage quality metric to determine the coverage quality of a cell given a set of camera configurations by modeling realistic camera characteristics. A cell was defined as any unit of observable space, like a square in a grid or a triangle in a triangular mesh. Furthermore, they proposed a cost function that combines this metric with observed human occupancy for optimization. We extend this notion and define the Camera Coverage Quality Metric (CCQM) to quantify amount of observable space (A), amount view of regions with expected dominant activity (H), amount of ability to capture the preferred pose (F) and image resolution of these objects (R) for a camera configuration ω. The Camera Coverage Quality Metric ($CCQM$) is defined as

$$CCQM(\omega) = g(A, H, F, R) = A(\omega) * H(\omega) * F(\omega) * R(\omega) \tag{4.5}$$

The optimal configuration of the cameras in G is defined as

$$\{\omega_1^*, \omega_2^*, ..., \omega_\nu^*\} = \underset{\{\omega_1, \omega_2, ..., \omega_\nu\} \in \Omega}{\arg\max} \sum_{i=1}^{\nu} CCQM(\omega_i) \tag{4.6}$$

Given ω the functions $\{A, H, F, R\}$ are defined as follows. Without loss of generality we assume that the geometry to be viewed is represented by a triangular mesh containing triangles $\{t_1, t_2, ...t_n\}$ with centroids $\{c_1, c_2, ...c_n\}$. Let $\{t_1^\omega, t_2^\omega, ...t_m^\omega\}$ be the set of triangles in view of the camera with configuration ω.

Amount of observable space: The geometric area in view of the camera is used to quantify this aspect. The area of coverage function $A(\omega)$ is defined as

$$A(\omega) = \frac{area\_in\_view}{total\_area} = \frac{\sum_{i=1}^{m} area(t_i^{\omega})}{\sum_{i=1}^{n} area(t_i)} \tag{4.7}$$

4.2.2.1 Amount of view of regions with expected dominant activity

An occupancy map of a space quantifies how often a point is accessed compared to other points in that space. Let us assume an occupancy map as defined in [65], that defines the frequency with which a triangle is accessed by humans. The same methodology as followed in [65] is used to compute the occupancy map. The amount of occupancy is used to define the activity in the area. If $O(t)$ is the occupancy of the triangle t, then the human occupancy volume function is defined as

$$H(\omega) = \frac{\sum_{i=1}^{m} O(t_i^{\omega})}{\sum_{i=1}^{n} O(t_i)} \tag{4.8}$$

4.2.2.2 Amount of ability to capture the preferred pose of objects

Humans are considered as objects of interest. Assuming that $\tau = \{T_1, T_2, ...\}$ be a set of trajectories followed by humans in the geometry G. These trajectories are used to quantify the amount of frontal view that can be obtained from the configuration ω. For every triangle t_i in the floor triangular mesh, direction discretization is performed

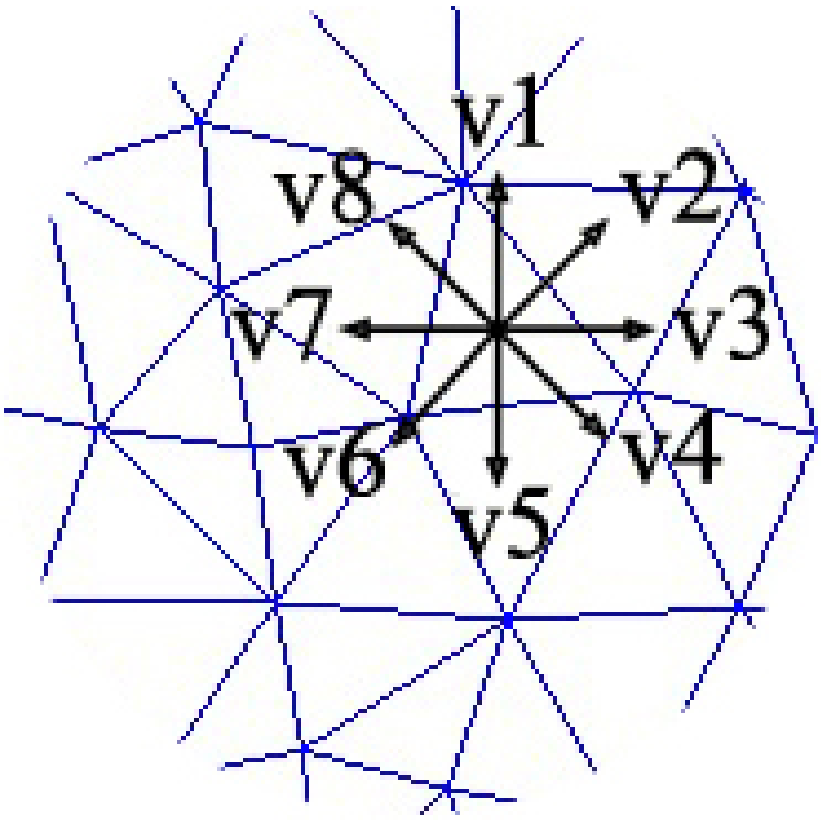


Figure 4.1: Vector discretization of triangle in a triangular mesh for creating a vector transition histogram from trajectories. .

and eight direction vectors $\{v_1^i, v_2^i, ..., v_8^i\}$ are defined as in [111] by Zhou *et al.* See Figure 4.1.

In the following step, a vector transition histogram is constructed from the set of these trajectories. Consecutive points in the trajectory are considered to create a direction vector. If $T = \{p_1, p_2, ..., p_l\}$ is a trajectory of length l, for all set of consecutive points $\{p_{i-1}, p_i\}$, the direction vector is defined as $(p_i - p_{i-1})$. The bin corresponding to the triangle t in which the point p_{i-1} is located and the discretized

direction vector subtending the smallest angle with $(p_i - p_{i-1})$ is incremented. Let $\Psi(t, v) \mapsto \mathbb{R}$ where $t \in \{t_1, t_2, ..., t_n\}$ and $v \in \{v_1, v_2, ..., v_8\}$ be the histogram function, then the frontal pose function $F(\omega)$ for a camera with center C is defined as

$$\begin{aligned} F(\omega) = \frac{1}{m} \sum_{i=1}^{m} (((C - c_i) \cdot v_{k-1}) \Psi(t_i, v_{k-1}) \\ + ((C - c_i) \cdot v_k) \Psi(t_i, v_k) \\ + ((C - c_i) \cdot v_{k+1}) \Psi(t_i, v_{k+1})) \end{aligned} \tag{4.9}$$

$$k = \arg\max_k (v_k \cdot (C - c_i)) \tag{4.10}$$

where t_i is the triangle with centroid c_i and v_k is the direction vector that subtends the smallest angle with $(C - c_i)$.

4.2.2.3 Image resolution of the object

This component of $CCQM$ quantifies the resolution of the face. If the obtained image is far from the camera, the obtained resolution is very low and the image might not add any value to the system. This component is application dependent, it could be customized to obtain a sufficient resolution of any object, which could be just the face or the entire body of a human. We follow the method described by Janoos *et al.* [49] and define the function $R(\omega)$ for a camera with center C as

$$R(\omega) = \frac{1}{m} \sum_{i=1}^{m} \frac{\rho^{\omega}(t_i)}{\rho_{min}} \tag{4.11}$$

$$\begin{aligned}\rho^{\omega}(t_i) = (2\pi * d(C, c_i)^2(1 - \cos(\gamma/2)))^{-1} \\ (\sigma_{k-1}(C - c_i) \cdot v_{k-1} \\ +\sigma_k(C - c_i) \cdot v_k \\ +\sigma_{k+1}(C - c_i) \cdot v_{k+1})\end{aligned} \tag{4.12}$$

where γ is the Y-field of view defined for the camera, $d(p_1, p_2)$ is the Euclidean distance between the points p_1 and p_2, k is as defined in Equation 4.10, σ is the number pixels the object occupies in the image and ρ_{min} is the user defined value that defines a minimum required resolution of an object in $pixels/inch$.

4.2.3 Optimization

Now that a metric is defined to assess the quality of a camera configuration ω, we perform a search in the geometry G to find the optimum parameter ω^*. Given the geometry and the domain knowledge, the search is performed to find two points, first on the ceiling to position the camera and the second on the floor to point the camera towards. Hence the parameter ω contains a pair of 3D points $\{v_1, v_2\}$. A variation of the hill climbing algorithm called the random-restart hill climbing (RRHC) algorithm is used for finding the optimum parameter ω^*. Random-restart hill climbing is an optimization search that provides near optimal performance [109, 30]. The idea is to search a limited number of points randomly and choose the best start location for hill climbing optimization. Since the objective is to find two points, one on the floor and the second on the ceiling, this is done at two levels.

4.2.3.1 Optimal pair

This algorithm takes as input a point on the ceiling (v_1) along with the list of points on the floor as input and performs RRHC optimization to find the optimal pair v_2 (a point on the floor) for v_1 that maximizes CCQM. See Algorithm 2.

4.2.3.2 RRHC optimization

This algorithm takes as input a list of points representing the ceiling (C) and another list representing the points on the floor (F) and performs RRHC to find the optimal parameters $\{v_1, v_2\}$ that maximizes CCQM for a camera, where v_1 is a point to position the camera and v_2 is a point for orienting the camera towards. See Algorithm 3.

4.2.4 Framework

The framework for obtaining the optimal parameters $\{\omega_1^*, \omega_2^*, ..., \omega_\nu^*\}$ given the geometry G is described in this section. The framework design is shown in Figure 4.2 which contains three modules.

1. **Model:** In this module, the infrastructure is modeled. This requires domain knowledge regarding the infrastructure such as entrances, exits and doors (nodes). Furthermore, knowledge regarding the frequency of accessing these nodes is also required. The output is a list of transitions between nodes.

2. **Data generation:** In this module, the data required for optimization is generated. The input is the list of node transitions from the previous module. First

Algorithm 2 Optimal Pair

Require: v_1 (ceiling point), L (floor points list)
Ensure: v_2 (Optimal floor point)

1: **procedure** OPTIMAL–PAIR
2: //Random Search
3: $n \leftarrow$ number of points for random search
4: $currentv_2 \leftarrow Random\_Solution(L)$
5: $current \leftarrow CCQM(v_1, currentv_2)$
6: **for** $(i = 1; i \leq n; i++)$ **do**
7: $currentv_2 \leftarrow Random\_Solution(L)$
8: $candidate \leftarrow CCQM(v_1, currentv_2)$
9: **if** $candidate > current$ **then**
10: $current \leftarrow candidate$
11: $candidatev_2 \leftarrow currentv_2$
12: **end if**
13: **end for**
14: //Hill Climbing
15: $current \leftarrow CCQM(v_1, candidatev_2)$
16: **for** $k \in neighbors(candidatev_2)$ **do**
17: $currentv_2 \leftarrow candidatev_2.neighbor[k]$
18: $candidate \leftarrow CCQM(v_1, currentv_2)$
19: **if** $candidate > current$ **then**
20: $current \leftarrow candidate$
21: $v_2 \leftarrow currentv_2$
22: **end if**
23: **end for**
24: $Return(v_2)$
25: **end procedure**

Algorithm 3 RRHC Optimization

Require: C (ceiling points list), F (floor points list)
Ensure: v_1, v_2 (Optimal pair)

procedure RRHC–Optimization
    //Random Search
    $n \leftarrow$ number of points for random search
    $currentv_1 \leftarrow Random\text{–}Solution(C)$
    $currentv_2 \leftarrow Optimal\text{–}Pair(currentv_1)$
    $current \leftarrow CCQM(currentv_1, currentv_2)$
    for $(i = 1; i \leq n; i++)$ **do**
        $candv_1 \leftarrow Random\text{–}Solution(C)$
        $candv_2 \leftarrow Optimal\text{–}Pair(candv_1)$
        $candidate \leftarrow CCQM(candv_1, candv_2)$
        if $candidate > current$ **then**
            $Maxv_1 \leftarrow candv_1$
            $current \leftarrow candidate$
        end if
    end for
    //Hill Climbing
    $currentv_1 \leftarrow Maxv_1$
    $currentv_2 \leftarrow Optimal\text{–}Pair(currentv_1)$
    $current \leftarrow CCQM(currentv_1, currentv_2)$
    for $k \in neighbors(currentv_1)$ **do**
        $candv_1 \leftarrow currentv_1.neighbor(k)$
        $candv_2 \leftarrow Optimal\text{–}Pair(candv_1)$
        $candidate \leftarrow CCQM(candv_1, candv_2)$
        if $candidate > current$ **then**
            $current \leftarrow candidate$
            $v_1 \leftarrow candv_1$
            $v_2 \leftarrow candv_2$
        end if
    end for
    $Return(v_1, v_2)$
end procedure

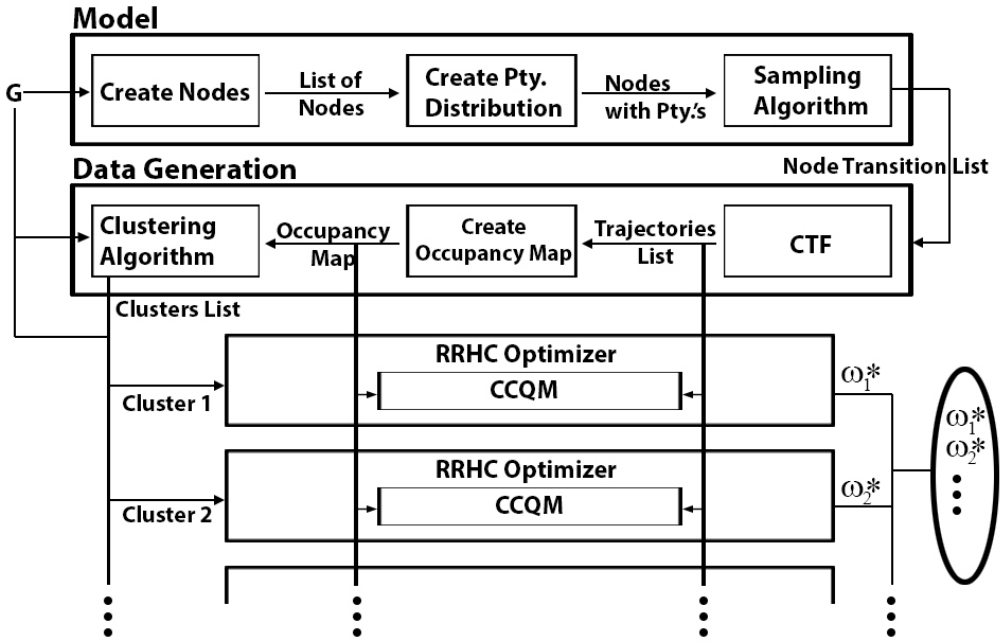


Figure 4.2: Framework with three modules, model, data generation, and RRHC optimizer for obtaining the optimal parameters $\{\omega_1^*, \omega_2^*, ..., \omega_\nu^*\}$.

a list of trajectories are generated using CTF for each pair of nodes from the list. These are the list of trajectories described in section 4.2.2 for quantifying the amount of preferred pose of objects of interest. These trajectories are then given as input to a sub-module that accumulates the trajectories to create an occupancy map that describes the frequency with which humans access the geometry. This occupancy map is the function $O(t)$ described in section 4.2.2 for quantifying the amount of view of regions with dominant activity. Then the occupancy map is also input to a clustering algorithm to cluster points based on their occupancy and spacial location in the geometry.

3. **RRHC optimizer:** Each one of these clusters obtained is given as input to optimizers for finding the optimized configuration $\{\omega_1^*, \omega_2^*, ..., \omega_\nu^*\}$ for each cluster.

4.3 People Tracking

This proposed human trajectory prediction method does not account for interaction with dynamic objects (humans). For this application, the proposed method is enhanced to handle social interaction with humans and incorporated into a tracking algorithm.

Let G be the geometry of the environment and $P = \{p_1, p_2, ...\}$ be accessible points on the floor. Let there exists a function $F : P \rightarrow \Re$ that quantifies the accessibility $F(p)$ of a point p on the floor with respect to the geometry and other humans in the environment. Let $\tau = \{(p_1^\tau, t_1^\tau), (p_2^\tau, t_2^\tau), ..., (p_n^\tau, t_n^\tau)\}$ be human motion trajectory from p_1^τ to p_n^τ such that $(p_1^\tau, ..., p_n^\tau) \in P$ and t_i^τ is the time stamp when the human is located at p_i^τ. Given the function F, the trajectory can be modeled as a Markov chain model.

$$P(p_{i+1}|p_i, p_{i-1}, ..., p_1, F) = P(p_{i+1}|p_i, F) \tag{4.13}$$

Given this probability distribution, consecutive points can be sampled from the floor to form a trajectory. The problem then simplifies to generating a function F that accounts for destination, geometry and other humans in the environment and assign values higher values to the points on the floor that adhere to social norms. The

original method is recapped to describe how the third component that models the effect of dynamic objects influences the occupancy map.

Consider the geometry shown in figure 4.3. The effect of the factors (distance, geometry and humans) on this floor plan are demonstrated below.

Figure 4.3: Geometry of a Floor Plan.

4.3.1 Destination

Consider the destination a human is trying to reach be as shown in figure 4.4. A distance map $D(p_i)$ is created which indicates the distance of the point p_i from the destination. In the absence of the effect of the geometry or other human in the environment, one could create a probability distribution that is inversely proportional to the distance map and sample points consecutively to generate a trajectory to the destination. This would result in a trajectory that represents the shortest path to the destination.

$$F(p_i) \propto -D(p_i) \tag{4.14}$$

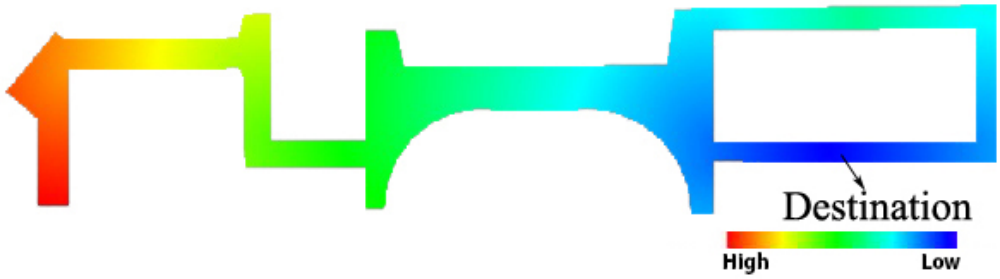


Figure 4.4: Distance Map to Destination.

4.3.2 Geometry

Given the geometry of the environment and static objects in it, hypothetically if a large number of trajectories followed by humans are observed in the environment, it can be assumed that certain points would be accessed more often than the other. For example, points on the floor that are next to the wall or an object might be accessed less often than those that are farther away from any static geometry. As describe in the forecasting algorithm the obtained accessibility map is shown in figure 4.5.

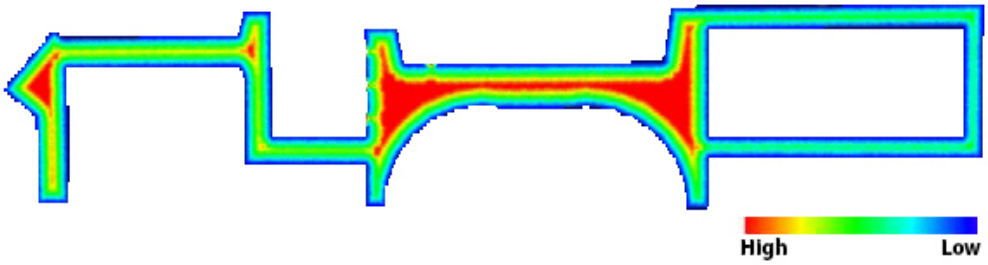


Figure 4.5: Accessibility Map based on the geometry.

The accessibility map was combined with the distance map (Figure 4.6) to obtain a distribution that allows sampling points for the trajectory that represent the

shortest distance while following social norm concerning geometry and objects. The function F for any point p_i in the geometry was defined as

$$F(p_i) = -D(p_i)/A(p_i) \tag{4.15}$$

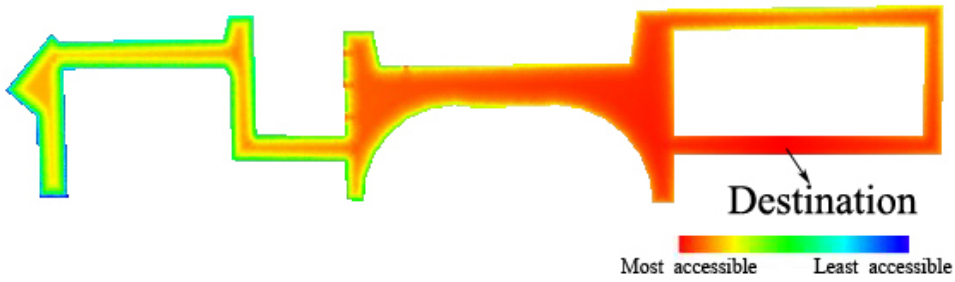


Figure 4.6: Accessibility Map Combined with Distance Map to Destination.

4.3.3 Humans

Objects and geometry are static and so is their effect on the accessibility map and human motion. However, humans in the environment are dynamic and hence their effect on the accessibility of a point is also dynamic. A human motion is effected by the other humans in the environment and vice versa. The effect of other humans on the accessibility map is modeled using the Theory of Proxemics [39]. Theory of Proxemics in an observational study, that define how humans utilize the physical space around them. This theory classifies the space close to a human into four discrete regions,: Intimate, Personal, Social and Public distance. The proposed method adapts a continuous effect on the accessibility map. Let a human be present

at the point p_i on the floor. The effect of this human on the accessibility map is defined as an exponentially increasing function with distance from the location of the human.

$$H(p_j) = 1 - \exp^{-d(p_i,p_j)/k} \tag{4.16}$$

Where, $d(p_i, p_j)$ is the Euclidean distance between the points p_i and p_j and k is a constant. This would make the accessibility at the location of the human ($p_i = p_j$) to be zero and increase exponentially as the distance increase. This is combined with the effect of the geometry and the destination to obtain F.

$$F(p_i) = \frac{-D(p_i)}{A(p_i)} \prod_j (1 - exp^{d(p_i,p_j)/k}) \tag{4.17}$$

where p_j is the position of the humans in the environment in view of the human whose motion is predicted. The obtained function F (shown in Figure 4.7) illustrates the effect of two human on the accessibility map. This would be the accessibility map for a third human trying to reach the destination.

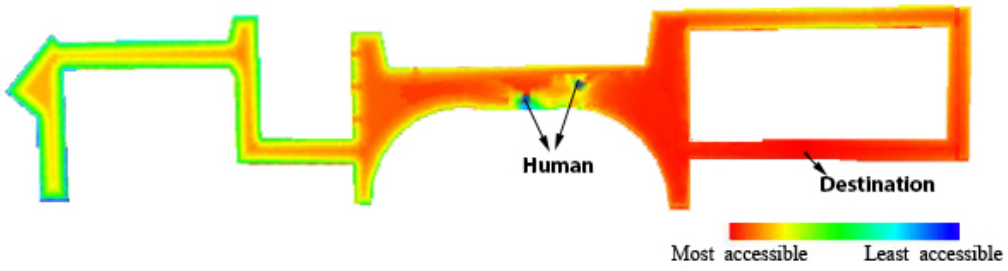


Figure 4.7: Effect of other humans on the Accessibility Map.

4.3.3.1 Trajectory Sampling

The function F is used to build a transition matrix for sampling points in the Markov chain. If the current location is p_t, we assume that the only possible points of transition are the neighbors of $\{p_{t1}, ..., p_{tm}\}$. The probability of transitioning to these neighbors is defined as

$$\propto \left\{ \begin{array}{ccc} F(p_{tm}) - F(p_t) & if & D(s_{tm}) - D(s_t) \leq 0 \\ 0 & otherwise & \end{array} \right\} \tag{4.18}$$

In order to reach the destination in the shortest time possible while conforming to the social norms, only the points closer to the destination are chosen ($D(p_{tm}) - D(p_t) <= 0$). The neighbors are sampled to form consecutive points in the trajectory.

4.3.4 Framework

The proposed tracking framework involves five steps as shown in figure 4.8.

1. Initialize the 3D model, occupancy map, location and appearance of the humans to be tracked.

2. Predict the future location of the human using CTF.

3. Localize a search region around the predicted locations and perform human detection.

4. Associate the observed data with the existing data using maximum likelihood - minimum meas square error filter based on the location and appearance.

5. Update the location and the histogram input to step 2 to continue prediction.

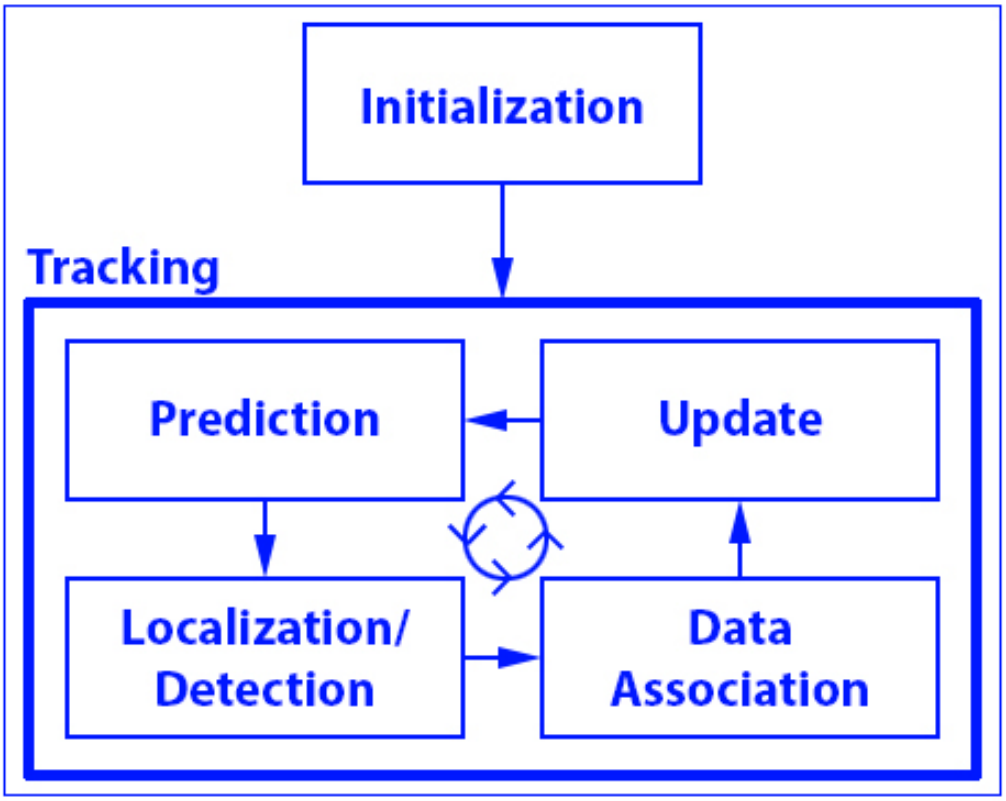


Figure 4.8: Tracking Framework.

Let $Y = \{y_1, y_2, ...\} = \{(p_{y_1}, h_{y_1}), (p_{y_2}, h_{y_2}), ...\}$ be the description of human $i = \{1, 2, ..., n\}$ being tracked, where p_i is the physical location in 3D geometry and h_i the HSV histogram of the human at time t_i. Given the geometry, occupancy map and the corresponding destination of the human, the future location of the human is predicted using CTF. Let p_i be the predicted location of the human at time t'_i. The point p'_i is projected on tho the image plane and a search region s'_i is defined. The search region is subjected to a human detection algorithm to obtain observations $Z = \{z_1, z_2, ...\} = \{(p_{z_1}, h_{z_1}), (p_{z_2}, h_{z_2}), ...\}$ where $j = \{1, 2, ..., m\}$ be the set of all

observations.

A maximum likelihood minimum mean square error data association filter is used to assign the observed data (Z) to the current state data (Y). Let $A_i = \{(y_{i_1}, z_{i_1}), (y_{i_2}, z_{i_2})...\}$ be an association such that $y_{i_j} \in Y, z_{i_j} \in Z$ and $A_i \in A$, where A is the set of all mutually exclusive and exhaustive events between the sets Y and Z.

$$\begin{aligned} i &= (y, z) \\ &= \arg \max_{y \in Y, z \in Z} P(y, z|A_i) \end{aligned} \tag{4.19}$$

$$\begin{aligned} P(z, y|A_i) &= P(z, y|A) \\ &= P(z = z_i|y = y_i) \\ &= P((p_{z_i}, h_{z_i})|(p_{y_i}, h_{y_i})) \\ &= P(p_{z_i}|p_{y_i}) * P(h_{z_i}|h_{y_i}) \end{aligned} \tag{4.20}$$

$$\begin{aligned} P(p_{z_i}|p_{y_i}) &\propto (1 - d(p_{z_i}, p_{y_i})) \\ P(h_{z_i}|h_{p_i}) &\propto d_h(h_{z_i}, h_{y_i})) \end{aligned} \tag{4.21}$$

Where $d(p_{z_i}, p_{y_i})$ is the Euclidean distance between the point p_{z_i} and p_{y_i} and $d_h(h_{z_i}, h_{y_i})$ is the histogram intersection distance between h_{z_i} and h_{y_i}. Finally, the corresponding state of the human are updated according to the association model.

Chapter 5

Implementation

5.1 Re-Identification and People Tracking

The implementation is similar for both the applications, re-identification and people tracking. This section describes how a complete 3D model of the environment can be constructed, and furthermore how cameras in the real world are calibrated and then embedded as virtual cameras in the model.

5.1.1 Modeling 3D environment

The 3D geometry of the environment like floors, walls, hallways, etc. are modeled using Google Sketchup, a 3D modeling tool. Figure 5.1 depicts the 3D model of a

building constructed using existing floor plans to obtain the measurements and dimensions. The 3D model is then exported using a *common digital asset exchange format* [3] called COLLADA file format. COLLADA Document Object Model (DOM) library is used to load and save this 3D model into an application, and then OpenGL is used to interact with this 3D data in the application.

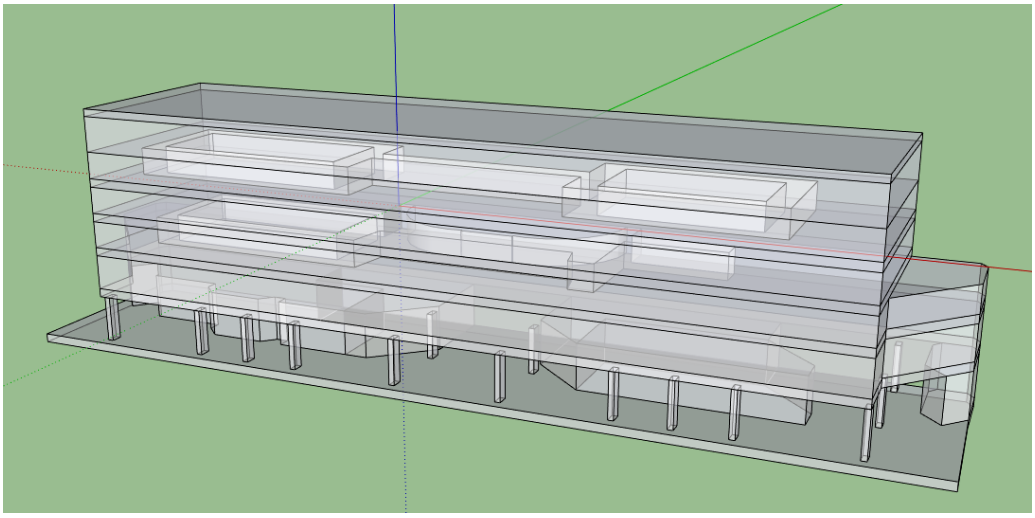


Figure 5.1: Model of a building using Google Sketchup.

5.1.2 Embedding virtual cameras and calibration

To create virtual cameras in the 3D model that represent cameras in real world. First the internal camera parameters of the existing real world camera are determined by using a general calibration approach with a checkerboard. These parameters are used to create virtual cameras which render perspective projections of the 3D model that are conceptually equivalent to the real world cameras. Now in order to determine the location and orientation of the camera in the 3D model, the image

from the real-world camera and manually registered with the corresponding camera's perspective projection in the 3D model, by manually changing the parameters in the transformation matrix using OpenGL. When the images register as shown in Figure 5.2, the transformation matrix of the camera is extracted which gives us the approximate location and orientation of the camera in the 3D model [86].

5.1.3 Delaunay triangulation of the floor mesh

The ground plane is represented as a triangular mesh though other representation are possible. Delaunay triangulation is used to obtain a uniform triangular mesh as shown in Figure 5.2. An implementation of the Delaunay triangulation is available in the Computational Geometric Algorithms Library (CGAL) [1]. The centroids of the triangles are considered as points on the ground plane.

5.1.4 Projecting points on the image into the 3D geometric model:

Human subjects captured from videos are projected into the ground plane in the 3D model, to obtain their global position. The location and orientation of the camera is available in the transformation matrix. The point on the image where the humans feet touches the ground plane is located and using the cameras parameters are projected on the ground plane.

5.2 Camera Placement Optimization

5.2.1 Model

Given the geometry of an infrastructure, most humans follow trajectories with a goal of reaching a destination like an entrance, exit or a doorway. There is a certain probability associated with accessing these nodes based on the purpose they serve in the infrastructure. For example at an airport, passengers might access the ticket counter with a higher probability than a coffee shop or a restroom. The knowledge of this probability can be used to sample nodes that humans can transition between. Let us consider the following test case scenario. In Figure 5.3, the objective was to install a network of cameras that provide effective surveillance in the hallway.

5.2.1.1 Create nodes and probability distribution:

The identified nodes are labeled with numbers in Figure 5.3. Let $\{n_1, n_2, ...\}$ be the nodes in the geometry G. In the absence of any observations of human motion, the probability of accessing a node was assumed to be proportional to the accommodation capacity of the room unless it was an entrance or exit. Implying that higher the capacity of a room to hold/seat people, the higher was the probability of accessing it. If $P_a(n_i)$ is a probability function that assigns probability to a node n_i and $A_c(n_i)$ is its accommodation capacity, then

5.2.1.2 Sampling algorithm:

The sampling algorithm was designed based on few assumptions. A human entering the geometry G would eventually exit. A human would access a minimum of one node before exiting the geometry. Algorithm 4 describes the steps.

Algorithm 4 Nodes Sampling

1: Choose a random entrance
2: Choose a node to access using P_a as distribution
3: Choose randomly to either exit or access another node
4: **if** access another node **then**
5: Choose another node excluding the current node
6: Goto step 3
7: **else**
8: Choose a random exit
9: **end if**

In the example geometry in Figure 5.3, an entry $(4, 7)$ was chosen with equal probability, then a node was chosen that is not an exit based on the assigned probability (P_a). Now assuming that the human had transitioned to the node, the human could either choose to transition to another node or exit with equal probability. If the human chose to exit, the closest exit was chosen, else the human would choose to go to another node based on a calculated probability. The probability of choosing the second node changed because the node that the human was currently in was eliminated when calculating the probabilities. This gave a list of nodes $\{n_1^s, n_2^s, ...\}$ that can be used as start and end nodes for simulating trajectories.

5.2.2 Data Generation

Given the geometry of the environment along with the nodes and their assigned probabilities, the likely human motion in the infrastructure was simulated to identify regions of dominant human activity.

5.2.2.1 Contextual trajectory forecasting (CTF)

CTF [65] was used to simulate trajectories from the start node to the end node. Given the 3D geometry of the environment and the starting point and destination of a human, CTF is assembled on two assumptions. First, the human would follow a path that requires the shortest time to reach the destination, and second, the human would adhere to certain behavioral norms that are observed when walking in hallways. CTF uses a Markov model and assigns probabilities to points on the floor such that consecutive points are sampled from start to destination to form a trajectory that represents the shortest path while conforming to observed behavioral norms. CTF can take any pair of nodes $\{n_i^s, n_j^s\}$ from the previous step and produce a trajectory $T_{ij}^s = \{n_i^s, p_1^s, p_2^s, ..., n_j^s\}$.

5.2.2.2 Create occupancy map ($O(t)$)

In this step, multiple pairs of nodes were generated as described in the previous step. These generated nodes were input to CTF to obtain a set of trajectories $\tau = \{T_1, T_2, ..., \}$. These are the set of trajectories used for quantifying the preferred pose of objects of interest as described in section 4.2.2. These trajectories were mapped

No.	Cluster	Occupancy
1	Blue	0.23
2	Red	**0.42**
3	Green	0.13
4	Aqua	0.11
5	Light Pink	0
6	Pink	0.11

Table 5.1: Identified clusters and their mean occupancies.

to the floor in the geometry to create an occupancy map $O(t_i)$ which quantifies the number of times a trajectory passes through a triangle t_i as used in quantifying the amount of view of regions with dominant activity in section 4.2.2. A snapshot of the occupancy map from the simulated trajectories T in G is shown in Figure 5.4 (a).

5.2.2.3 Clustering algorithm

The regions that belong to the same cluster should have a similar value of occupancy and also be located in the same spacial location. A point's spatial co-ordinates and it's occupancy $(c_i, O(t_i))$ were used as features, where $c_i = \{x_i, y_i, z_i\}$ are the 3D co-ordinates of the centroid of triangle t_i and $O(t_i)$ it's occupancy. The clusters obtained by using Expectation Maximization (EM) [26] are shown in Figure 5.4 (b). In this scenario, red cluster was identified to have the highest average human occupancy followed by blue and then pink as shown in Table 5.1.

5.2.3 RRHC Optimization

Once the clusters are identified, the optimization is applied on each cluster separately. Given a cluster, first the points in the ceiling that have a view of the centroid of the cluster are identified and these points are considered as the possible location of the cameras. The only possible orientation for a camera are pointing towards the points on the floor in the cluster. This would simplify the problem to finding two points, one on the ceiling to position the camera and the second on the floor to point the camera towards. As described in section 4.2.3, random restart hill climbing optimization was performed to find the two optimal points.

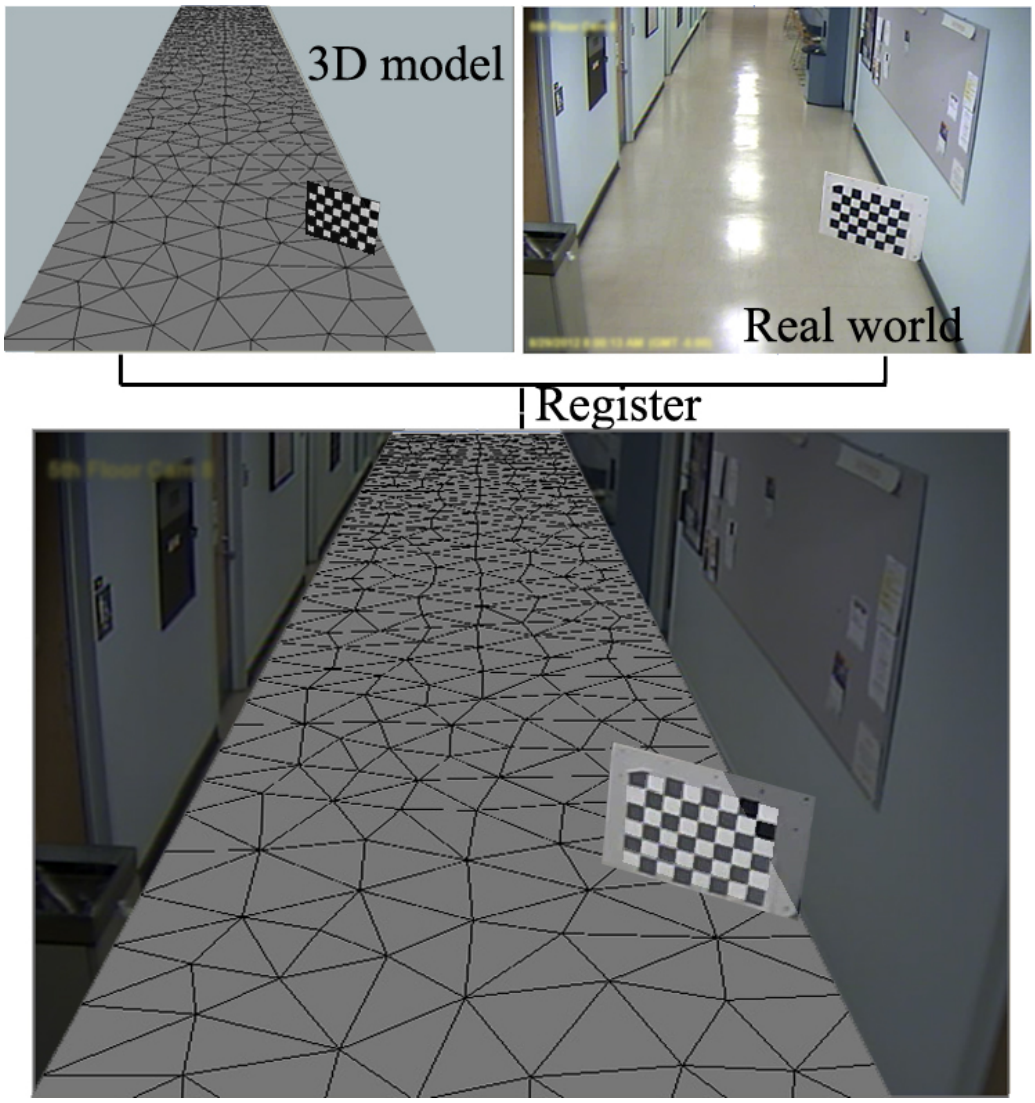


Figure 5.2: Manual registration of an image from a camera with the perspective rendering of the 3D model to extract the transformation matrix. The floor is represented by a uniform triangle mesh obtained by Delauney triangulation.

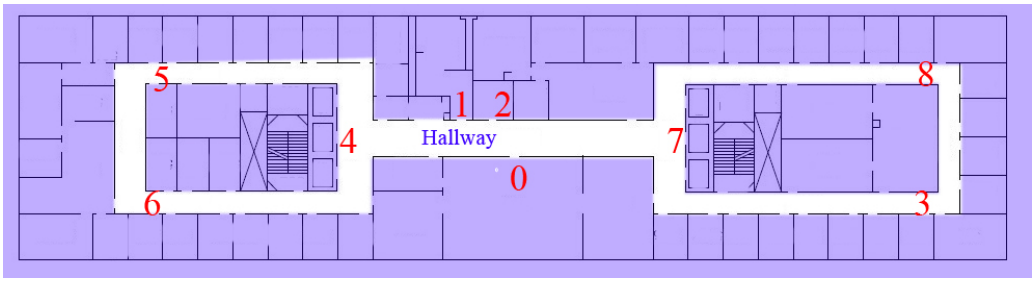
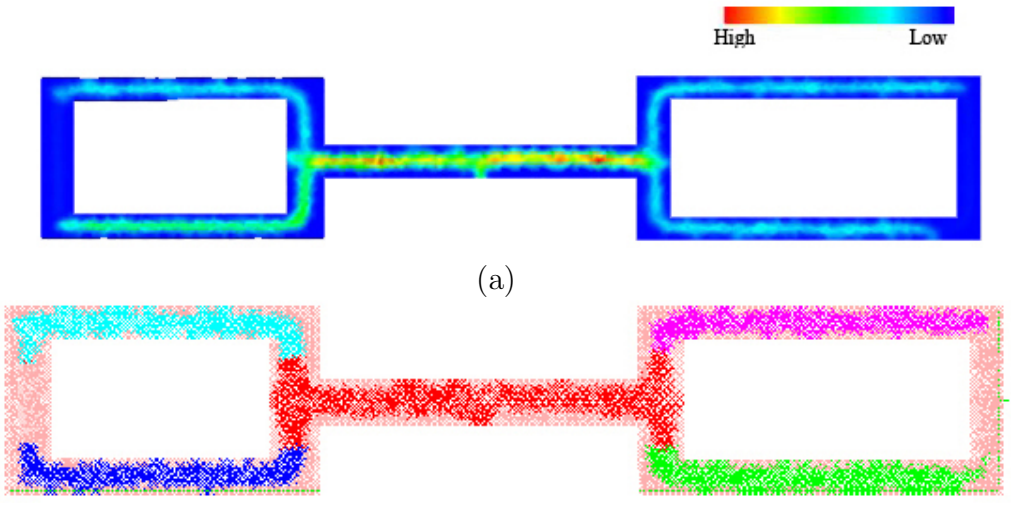


Figure 5.3: Floor plan of the test case scenario where the cameras are to be placed. The nodes are labeled with numbers.



(a)

(b)

Figure 5.4: (a) Occupancy map ($O(t)$) of the hallway obtained by mapping multiple simulated trajectories where red indicates regions of dominant activity and blue with minor activity, (b) Clusters of regions with dominant activity in the geometry obtained by EM algorithm.

Chapter 6

Experiments

6.1 Human Trajectory Forecasting

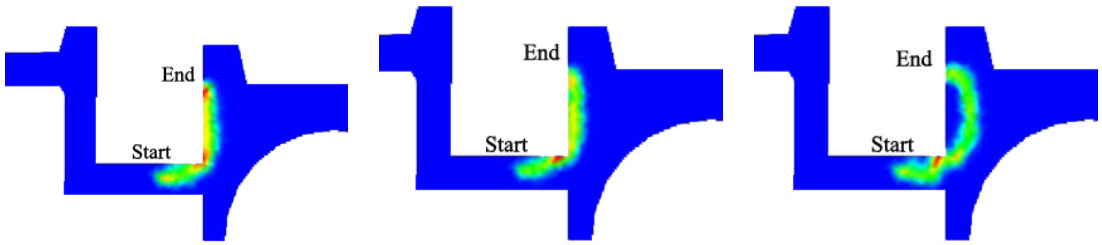
Real-world trajectories from three different scenarios were considered to evaluate the performance of the trajectory forecasting algorithm. The three scenarios are as shown in Figure 6.1. A sample size of 14 videos from scenario 1, 12 videos from

(a) (b) (c)

Figure 6.1: Experimental scenarios with a sample trajectory, red - actual trajectory, green - predicted trajectory; (a) scenario 1 (geometry A); (b) scenario 2 (geometry B); (c) scenario 3 (geometry B).

scenario 2 and 11 videos from scenario 3 (37 different human subjects) were used

to evaluate our trajectory forecasting model. None of these scenario's geometries were used in estimating the variable (β in linear regression or w and b in support vector machine regression) values during linear regression in Section 3.2. The human test subjects were given information regarding the destination only. An object was placed at the destination and all they were instructed was to walk to the destination, pick up the object and come back. The test subjects were not made aware that the purpose of the experiment was to observe their trajectories. The video of a test subject was taken and then processed through a human detection algorithm [24]. The detections were then projected into the 3D model to form a trajectory. Two different metrics were used to evaluate the model. In the first the modified Hausdorff distance was calculated between the real world trajectories and predicted trajectories. In the second the negative log-likelihood was determined for the real world trajectory by sampling it from a distribution created by the proposed model. The trajectory forecasting model was compared with a state-of-the-art approach (activity forecasting [55]) and a baseline algorithm. In the baseline algorithm, all the points are assumed to have equal occupancy hence allowing us to evaluate the impact of estimated model of human occupancy behavior. In activity forecasting [55], for each scenario the images are given semantic labels manually. To evaluate this approach on the dataset used for evaluation, the walls were labeled as building and the floor as sidewalk. The weights for the features/labels are learned from a different geometry and are then transfered and used for forecasting the trajectory distribution in the new geometry. Figure 6.2 compares the distribution of trajectories generated for scenario 1 using baseline, activity forecasting and proposed method.



(a) (b) (c)

Figure 6.2: Trajectory distribution around the corner for scenario 1 in geometry A; (a) Baseline; (b) Activity Forecasting; (c) Proposed Method;.

6.1.1 Modified Hausdorff distance:

Let $T_o = \{S_{o1}, S_{o2}, S_{o3}..\}$ be the observed trajectory and $T_q = \{S_{q1}, S_{q2}, S_{q3}..\}$ predicted trajectory, where $S_i \in P$ are points on the floor. The Hausdorff distance $D_H(T_o, T_p)$ between the two trajectories is defined as $\max\{D(T_o, T_q), D(T_q, T_o)\}$, where

$$\begin{aligned} D(T_o, T_q) &= \frac{1}{N_o} \sum_{a \in T_o} d(a, T_q) \\ d(a, T_q) &= \min_{b \in T_q} d(a, b) = \min_{b \in T_q} |a - b| \end{aligned} \tag{6.1}$$

$d(a, b)$ is the Euclidean distance between the points a and b, and N_o is the number of points in the trajectory T_o. This essentially is a metric for quantifying the difference between the trajectories T_o and T_q. Each trajectory is compared with 500 simulations of the predicted trajectory from our model to calculate the average modified Hausdorff distance. The average modified Hausdorff distance over the real world trajectories for the three geometries are shown in the Table 1. In all the three geometries, the error in predictions is decreased by using the distribution from the proposed model.

Scenario	Baseline	[55]	CFT-lr	CFT-sm
1	18.973	17.941	13.95	15.94
2	14.869	25.547	8.171	8.121
3	17.352	30.904	9.036	8.943

Table 6.1: Hausdorff distance of real world trajectories compared with simulated trajectories. The distances are measured in inches.

6.1.2 Log likelihood:

To directly compare our approach to the method in [55], given a starting point and a destination point, the proposed model was simulated multiple times and a transition probability matrix was constructed. If $T_q = \{S_{q1}, S_{q2}, S_{q3}..\}$ be the the simulated trajectory, using multiple simulations, a NXN transition matrix was constructed where N is the total number of states or points on the floor. Let $T_o = \{S_{o1}, S_{o2}, S_{o3}...\}$ be the observed trajectory. The probability of sampling the observed trajectory from the distribution created by the predicted trajectories was estimated as described in Activity Forecasting [55]. The models were simulated 2500 times to create the transition probability matrix. For an observed trajectory T_o, the error is estimated as

$$L(T_o) \propto E[\ln \prod_i P(S_{(i)o}|S_{(i-1)o})], \tag{6.2}$$

where $S_{(i-1)o}, S_{(i)o} \in T_o$ and $P(S_{(i)o}|S_{(i-1)o})$ is the probability of transition from state $S_{(i-1)o}$ (current triangle) to $S_{(i)o}$ (next triangle). This measure is normalized by dividing it with the length of the trajectory. The results in the Table 2 show the average negative log likelihood for each geometry. The results demonstrate how the

Scenario	Baseline	[55]	CFT-lr	CFT-sm
1	0.877	0.990	2.710	2.937
2	1.388	0.114	3.428	3.701
3	1.238	0.409	3.215	3.678

Table 6.2: Log likelihood of real world trajectories compared to simulated trajectories.

energy function decreases the error in forecasting the trajectory.

6.2 Person Re-Identification

Over the years many datasets like CAVIAR [6] and VIPeR [37] have been used for evaluating re-ID algorithms, but none of these datasets are equipped with the environments geometry and camera calibration. To evaluate the performance of the proposed method, real world data was collected from two different geometries. Each geometry consisted of three cameras with non-overlapping views in a hallway as shown in Fig. 6.3 (geometry A) and 6.4 (geometry B). Human subjects were allowed to walk down the hallway starting from camera 1 and are allowed to randomly choose between making either a left or right to show up in either camera 2 or 3 respectively. The images from camera 1 were used to create the gallery set and camera 2 and 3 were used to create the probe set. To simulate a real-world environment, groups of subjects were allowed to start walking at the same time from camera 1. The evaluation was performed on 38 subjects, 26 of which were used in geometry A and 12 in geometry B. In geometry A, 10 groups containing two subjects started at the same time and the rest started individually, and in geometry B, 2 groups of 4 subjects and 1 group of 3 subjects started at the same time and the rest individually. In both

the geometries, half of them were captured in camera 2 and the other half in camera 3 to create the probe set. For each ID, 5 shots were captured in all cameras. So the gallery set in geometry A consisted of 130 images and geometry B consisted of 60 images.

To perform re-identification for a given probe, we perform CTF from every image in the gallery. The starting position is determined by the position of the subject in the gallery and the end being determined by the position of the subject in the probe. These points are re reprojected into the 3D model as described in Chapter 5.

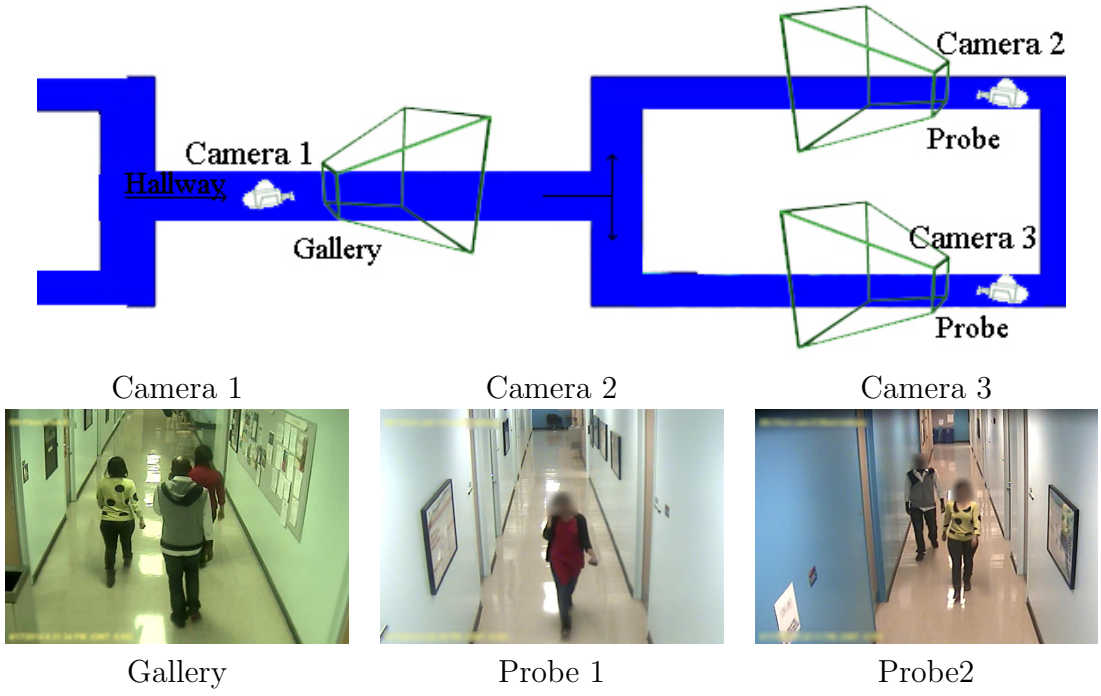


Figure 6.3: Geometry A experimental setup.

Experiments were performed in four different modes based on the number of shots used for calculating the scores. In single-shot vs single-shot (SvsS), each image in a set represented a different ID, in single-shot vs multiple-shot (SvsM), each image

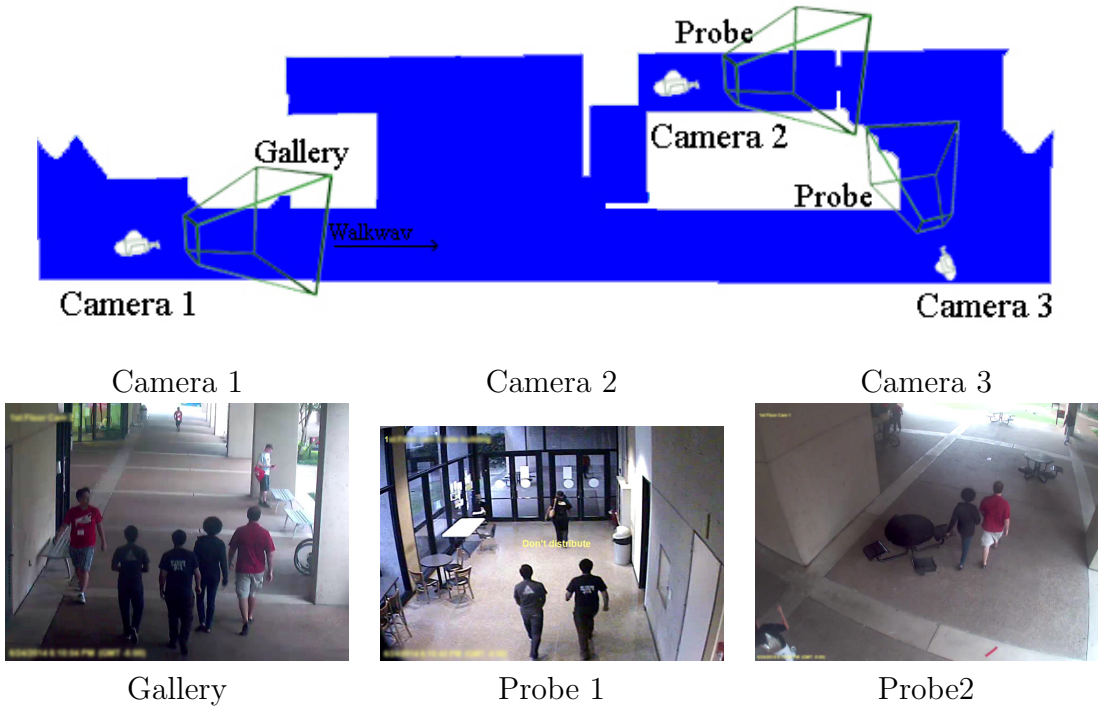


Figure 6.4: Geometry B experimental setup.

in gallery set is different ID but in the probe set, the scores from multiple shots of the same id were average out. In multiple-shot vs single-shot (MvsS), every shot in probe was compare to multiple shots belonging to the same ID in the gallery and the scores were averaged out, finally in multiple-shot vs multiple-shot (MvsM) multiple shots were used in both the gallery and probe set for matching. The results are presented in the form of recognition rate using Cumulative Matching Characteristic (CMC) curves.

In geometry A at most two subjects were allowed to start at the same time and hence a 100% recognition was obtained within the first two ranks in all the modes. Similarly in geometry B at most four subjects were allowed to start at the same time

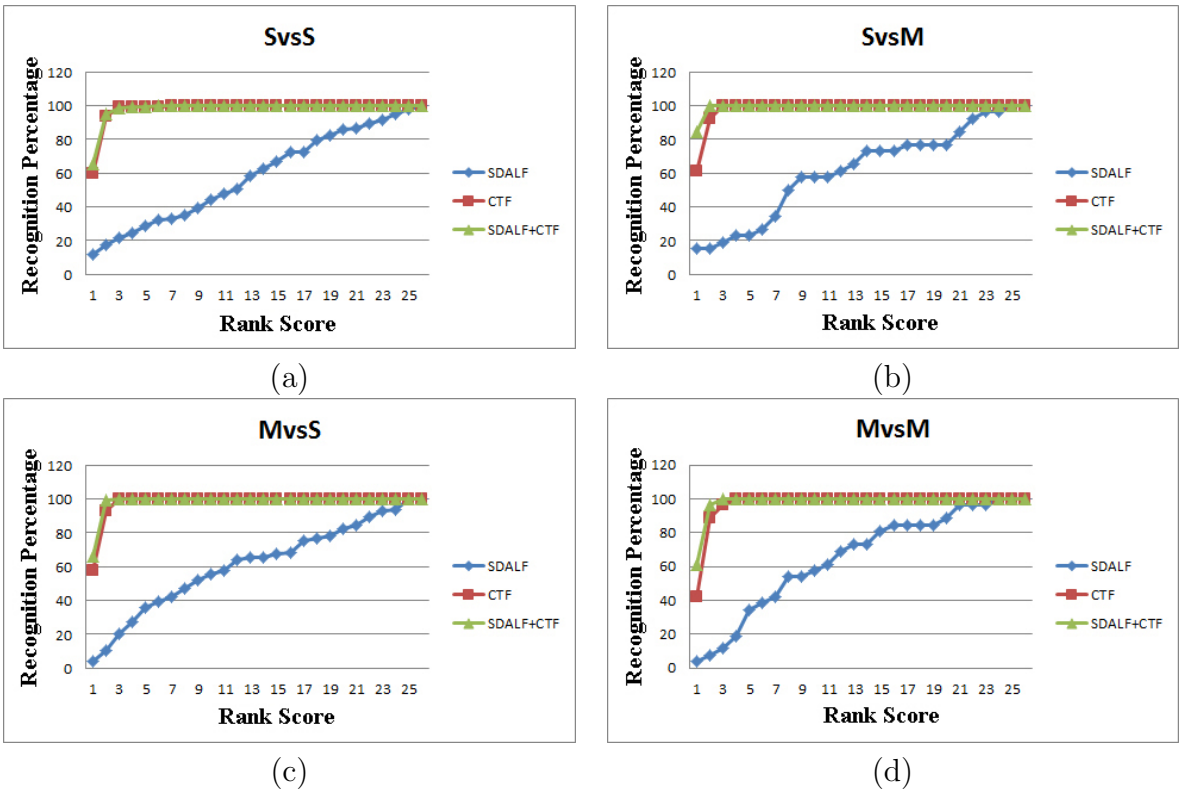


Figure 6.5: CMC curves: Geometry A

and hence a 95-100% recognition was obtained within the first four ranks in all the modes. In all the cases, in was observed that using CTF alone generated a significant boost in recognition over SDALF, and embedding CTF in SDALF generated a further enhancement in recognition performance over CTF.

6.3 Camera Placement Optimization

The motivation for this work was to optimize the camera placement in the geometry to provide effective surveillance as defined in section 1.2.2. A configuration of cameras

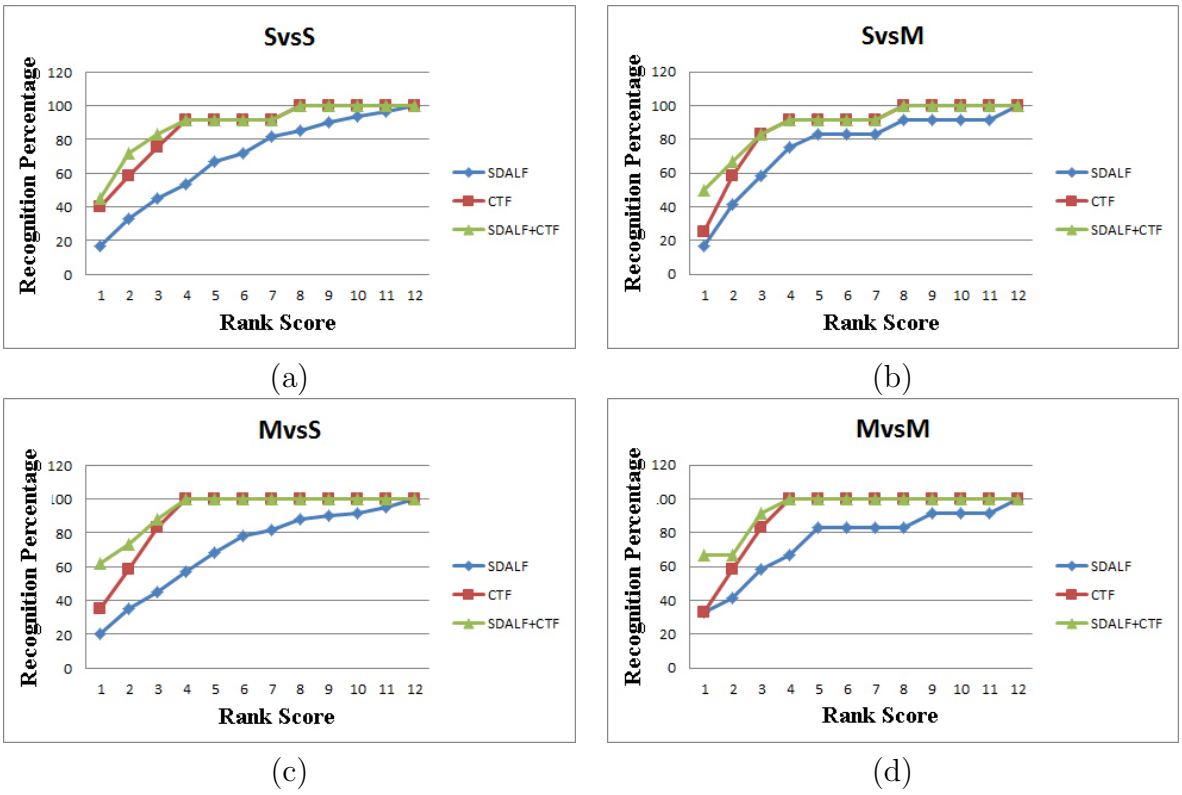


Figure 6.6: CMC curves: Geometry B

in a geometry is considered to provide effective surveillance if it maximizes the below quantities while minimizing the number of cameras. Such a system is effective both in terms of surveillance and cost. Hence all the quantities used for comparison are normalized by the number of cameras in the configuration.

- **Area of observable space in view**: The total area accessible by humans in view of the camera is calculated for all the cameras and normalized.
- **Amount of activity in view**: To quantify the occupancy of a location that is in view, the activity produced in that location is considered. The number of frames that have motion in them are used as a metric to define the activity of

the location that is viewed from the camera. The normalized value is used as a metric.

- **Pose of objects of interest and their resolution**: Assuming that a certain number of pixels are required for face detection. Face detection is used to quantify the pose of objects of interest along with their resolution. The number of faces detected are counted for every camera in the configuration and normalized.

The above metrics are defined to assess these qualities in a configuration of cameras. The configuration generated by the proposed method is compared to the following method.

- **3-coloring solution [31]:** A solution to Art Gallery Problem (AGP) was obtained using the 3-coloring solution and the cameras were placed at these locations. This configuration was used as baseline. The geometry of the environment's polygon contains holes. The polygon was modified to remove the holes and then 3 coloring solution was computed for the polygon. The cameras were manually placed to maximize the area in view. The solution is as shown in Figure 6.7 (a).

- **Janoos *et al.* [49]**: Janoos *et al.* defined cell coverage quality metric by taking observed human occupancy and resolution into account. This metric was used to optimize the camera location for each cluster. The following configuration was obtained, see Figure 6.7 (b).

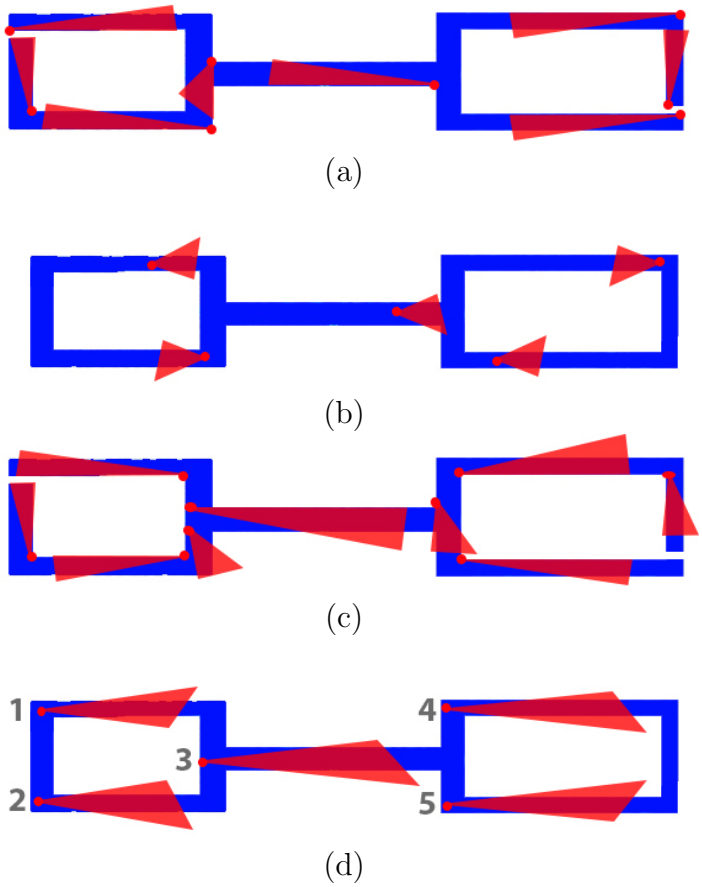


Figure 6.7: Configuration of cameras obtained from (a) 3 coloring solution, (b) Janoos *et al.*, (c) Huang *et al.*, (d) Proposed method.

- **Huang *et al.* [46]**: Huang *et al.* proposed a shortest watchman route solution and positioned wireless cameras along the route to maximize the view area of the polygon. Their solution was proposed only for simple polygons with out holes and hence the modified polygon was used in this case as well. The obtained configuration is shown in Figure 6.7 (c).

- **Proposed method:** The obtained configuration from the proposed method is

Method	no. of cams.	Area/cam	Activity/cam
3 Coloring	8	0.057	28048.6
Janoos	5	0.01	44366.2
Huang	10	0.064	40092.5
Proposed	5	**0.109**	**69933.8**

Table 6.3: Comparison of area and activity in view per camera.

shown in Figure 6.7 (d), and the view from the cameras are shown in Figure 6.8.

Cam. 1 Cam. 2 Cam. 3

Cam. 4 Cam. 5

Figure 6.8: Camera view from the cameras deployed in the test case scenario as calculated by the proposed method.

Table 6.3 shows the area under view per camera. Although, 3-coloring solution and Huang *et al.* has higher area coverage, the number of cameras used is higher than that of the proposed method and the area in view per camera is higher for the proposed method.

All cameras used for experiments had a frame rate of 30 fps. For each camera, the number of frames in which there is activity is counted using background subtraction. The average number of frames per camera are shown in Table 6.3. Most activity per

camera was observed in the proposed method.

For each of these methods, a day's worth of data (10 hours) was collected. We have run face detection [95, 58] on these videos to count the number of faces captured. The number of faces captured for each camera are shown in Table 6.4 (left). It can be noticed that Cam. 3 has the highest number of faces detected followed by Cam. 2. Cam. 3 is over-viewing the common hallway represented by the red cluster (see Table 5.1) with the highest simulated occupancy value. The average number of faces detected for each method are shown in Table 6.4 (right). Approximately the same total number of faces were detected by 3 coloring solution and the proposed method, except for 3-coloring solution uses 8 cameras and the proposed method uses only 5 cameras. Using Huang *et al.*, more than twice the total number of faces were detected than the proposed method but the number of cameras used were also twice as many than the proposed method. More than a quarter of the faces detected by Huang *et al.* configuration were from a single camera of the 10 cameras, which coincidentally happened to be focused at an elevator where people tend to stand and wait. The method proposed by Janoos *et al.* focuses on areas with high human occupancy and takes resolution of the triangle into account as opposed to the proposed method which uses the resolution of the approximate location of the face and hence their cameras are located above the regions of dominant human occupancy and fails to capture faces.

Although the proposed system performs better over the state of the art systems, some necessary improvements are to be taken into consideration. As noticed in Huang *et al.* configuration, significant number of faces were captured by focusing a camera

Camera	Faces
Cam. 1	622
Cam. 2	3430
Cam. 3	**5929**
Cam. 4	915
Cam. 5	1930

Method	cameras	Faces/cam
3 Coloring	8	1264
Janoos	5	1111.8
Huang	10	2040.5
Proposed	5	**2183.6**

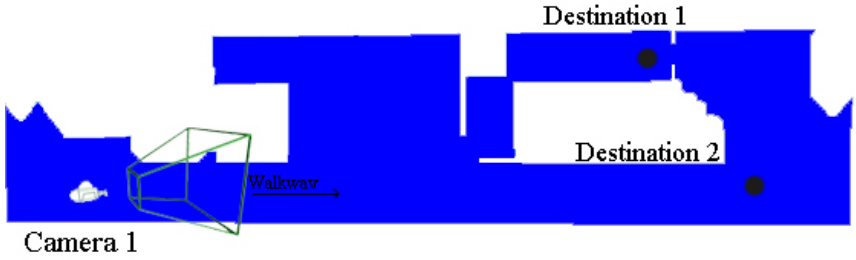
Table 6.4: (left) Faces counted from individual cameras in the proposed method, (right) Comparison of faces detected per camera.

at the elevator. This can be considered as a draw back of the proposed system and all the others being compared to, as none of the systems take the entrances and exits into consideration which could be valuable for surveillance. It would be interesting to incorporate a method to include the entrances and exits in the analysis. A method to estimate the number of cameras required for each cluster depending on the size of the cluster can be useful. If the cluster is big, it might be interesting to assign multiple cameras and incorporate a MCLP/BCLP problem formulation for optimization to ensure maximal coverage.

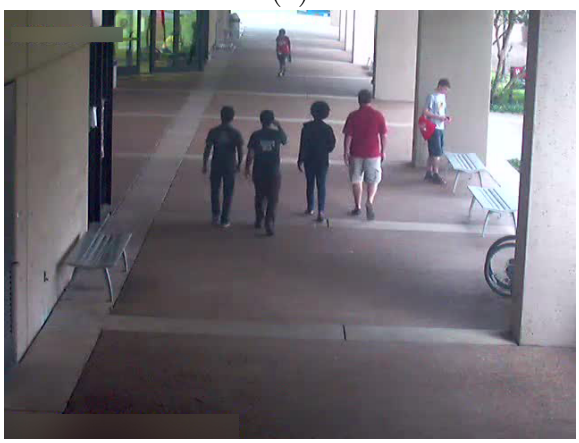
6.4 People Tracking

Two real-world scenarios were considered to evaluate the performance of the tracker. The geometry of the environment and their corresponding views are as shown in figure 6.9 and 6.10. A total of 49 ID's were used to evaluate the tracking algorithm of which 30 were from geometry B and the rest from geometry A. The dataset consisted of 15,000 frames containing 3 scenarios with 4 people, 5 scenarios with

3 people, 5 scenarios with 2 people walking simultaneously, the rest consisted of tracking 1 person.



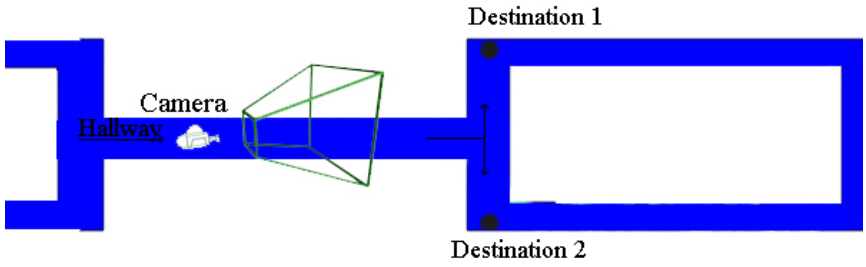
(a)



(b)

Figure 6.9: (a) Geometry A; (b) View of the camera located in Geometry A;

We compare the results of the proposed tracker against two other tracking algorithms. As a baseline, the tracker is first compared against the results from using histogram data alone from data association, i.e. with out the prediction. This will quantify the effect of the prediction algorithm on tracker's performance. Further more, the results are compared against [108], which is a state of the art online multi-person tracker proposed by Zhang *et al.* The results for geometry A are shown



(a)

(b)

Figure 6.10: (a) Geometry B; (b) View of the camera located in Geometry B

in figure 6.11, table 6.5 and for geometry B are shown in 6.12, table 6.6 and are quantified as misses, false positives and true positives.

The proposed algorithm has no misses, this is because, in the absence of a detection, the location of the object can be estimated from the trajectory prediction. The hierarchy tracker has the lowest number of false positives, this is because, if the detector fails to identify an object continuously, the algorithm stops tracking. Hence it has higher number of misses than the base line. The proposed algorithm has the highest number of true positives, out performing the baseline and the hierarchy

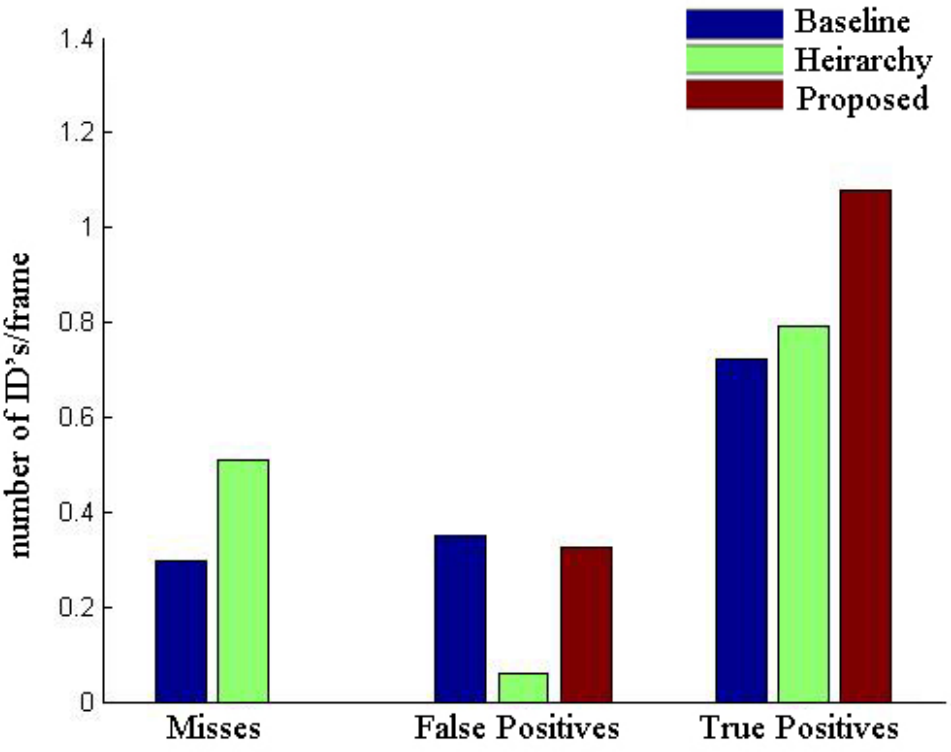


Figure 6.11: Misses, false positives and true positives for Geometry A tracker.

Method	Misses	False Positives	True Positives
Baseline	0.295	0.348	0.723
Zhang *et al.* [108]	0.509	0.0611	0.793
Proposed	0	0.325	1.076

Table 6.5: Misses, false positives and true positives shown as ID's per frame for Geometry A.

Method	Misses	False Positives	True Positives
Baseline	0.215	1.614	0.933
Zhang *et al.* [108]	1.284	0.172	1.174
Proposed	0	0.584	2.0252

Table 6.6: Misses, false positives and true positives shown as ID's per frame for Geometry B.

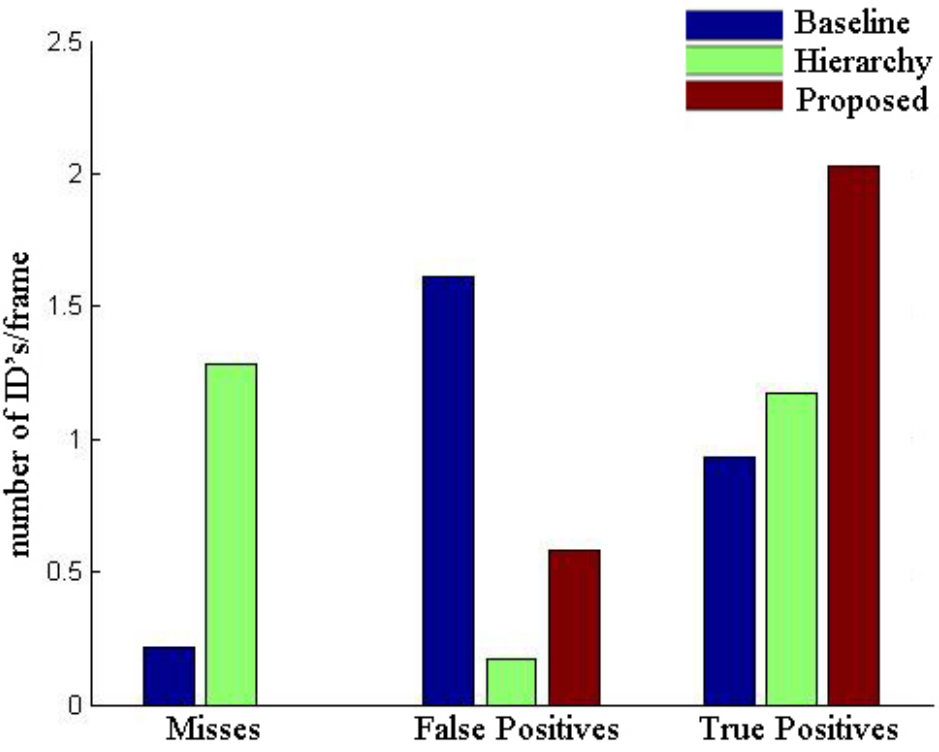


Figure 6.12: Misses, false positives and true positives for Geometry B

Chapter 7

Conclusion

We have modeled a set of geometric features that describe a point on the floor with respect to the structure of the surrounding geometry. We have proposed a method to estimate the occupancy map using the geometric features for any new geometry without the need for training data. We have developed an algorithm to forecast human motion trajectories using this estimated human behavior model.

We have successfully demonstrated the applicability of CTF in a traditional appearance based re-identification algorithm. We have proposed an algorithm to optimize the placement of surveillance cameras in a 3D infrastructure by predicting the possible human behavior within the infrastructure. We have proposed a method to identify regions with dominant human activity. We have also proposed a metric that quantifies the position of a camera based on the observable space, activity in this space, pose of objects of interest within the activity and their image resolution in camera view for optimization. Finally we have successfully demonstrated the

applicability of CTF into a multi-person tracking algorithm.

It is observed that incorporating the estimated occupancy map in the trajectory prediction can improve the accuracy of prediction significantly. The decrease in the log likelihood and the modified Hausdorff distance with the incorporation of the energy function supports the accuracy of this method. Preliminary results show that using the 3D geometry and contextual trajectory forecasting can enhance re-identification performance significantly over appearance methods. The proposed camera placement model was compared with the state of the art algorithms and the obtained results show an improvement in the amount of area under view, observed activity and face detection rate per camera. Preliminary results show that using the 3D geometry and contextual trajectory forecasting can enhance tracking performance significantly and the results were compared with the state of the art detection based tracking methods.

Bibliography

[1] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[2] G. Antonini, S. Martinez, M. Bierlaire, and J. Thiran. Behavioral priors for detection and tracking of pedestrians in video sequences. *International Journal of Computer Vision*, 2006.

[3] R. Arnaud and M. C. Barnes. *Collada: Sailing the Gulf of 3d Digital Content Creation*. AK Peters Ltd, 2006.

[4] K. Arras, S. Grzonka, M. Luber, and W. Burgard. Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008.

[5] G. Aschwanden, J. Halatsch, and G. Schmitt. Crowd simulation for urban planning. *Proceedings of eCAADe 2008*, 2008.

[6] M. Baeuml and R. Stiefelhagen. Evaluation of Local Features for Person Re-Identification in Image Sequences. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2011.

[7] S. Bak, E. Corvee, F. Bremond, and M. Thonnat. Person re-identification using spatial covariance regions of human body parts. In *Advanced Video and Signal Based Surveillance (AVSS)*, 2010.

[8] L. Bazzani, M. Cristani, and V. Murino. Symmetry-driven accumulation of local features for human characterization and re-identification. *Computer Vision and Image Understanding*, 2013.

[9] A. Bedagkar-Gala and S. K. Shah. A survey of approaches and trends in person re-identification. *Image and Vision Computing*, 2014.

[10] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun. Learning motion patterns of people for compliant robot motion. *Internationl Journal of Robotics Research*, 2005.

[11] S. Bhattacharya, V. Kumar, and M. Likhachev. Search-based path planning with homotopy class constraints. In *In Proc. National Conference on Artificial Intelligence*.

[12] P. Biber, H. Andreasson, T. Duckett, and A. Schilling. 3d modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 3430–3435 vol.4, Sept 2004.

[13] P. Biber, S. Fleck, and T. Duckett. 3d modeling of indoor environments for a robotic security guard. In *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pages 124–124, June 2005.

[14] P. Biber, S. Fleck, W. Strasser, S. Fleck, and W. Strasser. A mobile platform for acquisition of 3d-models of large environments : The wagele. In *In 3D-ARCH 2005: 3D Virtual Reconstruction and Visualization of Complex Architectures*, 2005.

[15] R. Bodor, A. Drenner, P. Schrater, and N. Papanikolopoulos. Optimal camera placement for automated surveillance tasks. *Journal of Intelligent and Robotic Systems*, 50(3):257–295, 2007.

[16] G. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, 1998.

[17] A. Bruce and G. Gordon. Better motion prediction for people-tracking. *Proc. of the Int. Conf. on Robotics and Automation ICRA*, 2004.

[18] Y.-y. Cao, G.-y. Xu, and T. Riegel. Moving object contour detection based on s-t characteristics in surveillance. In *Human Interface and the Management of Information. Methods, Techniques and Tools in Information Design*, Lecture Notes in Computer Science. 2007.

[19] X. Chen and J. Davis. Camera placement considering occlusion for robust motion capture. Technical report, 2000.

[20] X. Chen, K. Huang, and T. Tan. Learning the three factors of a non-overlapping multi-camera network topology. In C.-L. Liu, C. Zhang, and L. Wang, editors, *Pattern Recognition*, volume 321 of *Communications in Computer and Information Science*, pages 104–112. Springer Berlin Heidelberg, 2012.

[21] P. Y. Choi and M. Hebert. Learning and predicting moving object trajectory: a piecewise trajectory segment approach. Technical report, Robotics Institute, 2006.

[22] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2002.

[23] J. Cui, H. Zha, H. Zhao, and R. Shibasaki. Tracking multiple people using laser and vision. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005.

[24] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *ICCV*, June 2005.

[25] B. Debaque, R. Jedidi, and D. Prevost. Optimal video camera network deployment to support security monitoring. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 1730–1736, July 2009.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

[27] M. S. L. B. Dong Seon Cheng, Marco Cristani and V. Murino. Custom pictorial structures for re-identification. BMVA Press, 2011.

[28] T. J. Ellis, D. Makris, and J. K. Black. Learning a multi-camera topology. In *in Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 165–171, 2003.

[29] U. M. Erdem and S. Sclaroff. Optimal placement of cameras in floorplans to satisfy task requirements and cost constraints. In *In Proc. of OMNIVIS Workshop*, 2004.

[30] C. Filho, A. de Oliveira, and M. Costa. Using random restart hill climbing algorithm for minimization of component assembly time printed circuit boards. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 8(1):23–29, March 2010.

[31] S. Fisk. A short proof of Chvtal's Watchman Theorem. *Journal of Combinatorial Theory*, 24, 1978.

[32] S. Fleishman, D. Cohen-Or, and D. Lischinski. Automatic camera placement for image-based modeling. In *Computer Graphics and Applications, 1999. Proceedings. Seventh Pacific Conference on*, pages 12–20, 315, 1999.

[33] A. Fod, A. Howard, and M. J. Mataric. Laser-based people tracking. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3024–3029, 2002.

[34] S. Ghiasi, H. Moon, A. Nahapetian, and M. Sarrafzadeh. Collaborative and reconfigurable object tracking. *The Journal of Supercomputing*, 2004.

[35] H. Gong, J. Sim, M. Likhachev, and J. Shi. Multi-hypothesis motion planning for visual object tracking. In D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, editors, *ICCV*, pages 619–626. IEEE, 2011.

[36] H. Gong, J. Sim, M. Likhachev, and J. Shi. Multi-hypothesis motion planning for visual object tracking. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, 2011.

[37] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *Computer Vision ECCV 2008*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008.

[38] E. T. Hall. *The Hidden Dimension.* Anchor Books. ISBN 0-385-08476-5, 1966.

[39] E. T. Hall. *The Hidden Dimension.* Anchor Books. ISBN 0-385-08476-5, 1966.

[40] D. Helbing, L. Buzna, A. Johansson, and T. Werner. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation Science*, 39(1):1–24, 2005.

[41] D. Helbing and P. Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 1995.

[42] E. Horster and R. Lienhart. Approximating optimal visual sensor placement. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1257–1260, July 2006.

[43] E. Hörster and R. Lienhart. On the optimal placement of multiple visual sensors. In *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*, VSSN '06, pages 111–120, New York, NY, USA, 2006. ACM.

[44] E. Hrster and R. Lienhart. Calibrating and optimizing poses of visual sensors in distributed platforms. *Multimedia Systems*, 12(3):195–210, 2006.

[45] W. Hu, D. Xie, T. Tan, and S. Maybank. Learning activity patterns using fuzzy self-organizing neural network. *Sys., Man, and Cybernetics, IEEE Trans. on*, 2004.

[46] H. Huang, C.-C. Ni, X. Ban, J. Gao, A. Schneider, and S. Lin. Connected wireless camera network deployment with visibility coverage. In *INFOCOM, 2014 Proceedings IEEE*, pages 1204–1212, April 2014.

[47] S. Huang and J. Hong. Moving object tracking system based on camshift and kalman filter. In *Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on*, 2011.

[48] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Computer Vision ECCV '96*, Lecture Notes in Computer Science. 1996.

[49] F. Janoos, R. Machiraju, R. Parent, J. W. Davis, and A. Murray. Sensor configuration for coverage optimization for surveillance applications. In *SPIE, 2007 Proceedings*, 2007.

[50] O. Javed, K. Shafique, Z. Rasheed, and M. Shah. Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109(2):146 – 162, 2008.

[51] E. Jeon and S. Jo. Real-time building of a 3d model of an indoor environment with a mobile robot. In *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*, pages 818–823, Oct 2011.

[52] I. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. In *ICPR 2004*, 2004.

[53] K. Kim and A. T. Murray. Enhancing spatial representation in primary and secondary coverage location modeling*. *Journal of Regional Science*, 48(4):745–768, 2008.

[54] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert. Activity forecasting. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision ECCV 2012*, volume 7575 of *Lecture Notes in Computer Science*, pages 201–214. Springer Berlin Heidelberg, 2012.

[55] K. Kitani, B. D. Ziebart, J. A. D. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, Oct. 2012.

[56] R. Layne, T. Hospedales, and S. Gong. Towards person identification and re-identification with attributes. In *Computer Vision ECCV 2012. Workshops and Demonstrations*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012.

[57] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz. Voronoi tracking: location estimation using sparse and noisy sensor data. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2003.

[58] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900–I–903 vol.1, 2002.

[59] C. Loy, T. Xiang, and S. Gong. Time-delayed correlation analysis for multi-camera activity understanding. *International Journal of Computer Vision*, 90(1):106–129, 2010.

[60] C. Loy, T. Xiang, and S. Gong. Incremental activity modeling in multiple disjoint cameras. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1799–1813, Sept 2012.

[61] M. Luber, G. Diego Tipaldi, and K. O. Arras. Place-dependent people tracking. *Int. J. Rob. Res.*, 2011.

[62] M. Luber, J. Stork, G. Tipaldi, and K. Arras. People tracking with human motion predictions from social forces. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010.

[63] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–205–II–210 Vol.2, June 2004.

[64] R. Malik and P. Bajcsy. Automated Placement of Multiple Stereo Cameras. In *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS*, Marseille, France, Oct. 2008. Rahul Swaminathan and Vincenzo Caglioti and Antonis Argyros.

[65] P. Mantini and S. Shah. Human trajectory forecasting in indoor environments using geometric context. In *Proceedings of the Ninth Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP '14. ACM, 2014.

[66] D. Martínez, L. Velho, and P. C. Carvalho. Computing geodesics on triangular meshes. *Comp. Graph.*, Oct. 2005.

[67] R. Mazzon, S. F. Tahir, and A. Cavallaro. Person re-identification in crowd. *Pattern Recogn. Lett.*, 33(14):1828–1837, Oct. 2012.

[68] M. Mirabi and S. Javadi. People tracking in outdoor environment using kalman filter. In *Intelligent Systems, Modelling and Simulation (ISMS), 2012 Third International Conference on*, 2012.

[69] A. Mittal and L. Davis. Visibility analysis and sensor planning in dynamic environments. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 175–189. Springer Berlin Heidelberg, 2004.

[70] B. Morris and M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *CSVT*, 2008.

[71] A. T. Murray, K. Kim, J. W. Davis, R. Machiraju, and R. Parent. Coverage optimization to support security monitoring. *Computers, Environment and Urban Systems*, 31(2):133 – 147, 2007.

[72] J. Nascimento, M. Figueiredo, and J. Marques. On-line classification of human activities. In *Pattern Rec. and Image Analysis*, volume 4478 of *Lec. Notes in Comp Sci.* 2007.

[73] C.-J. Pai, H.-R. Tyan, Y.-M. Liang, H.-Y. M. Liao, and S.-W. Chen. Pedestrian detection and tracking at crossroads. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, 2003.

[74] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, 2009.

[75] B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by support vector ranking. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2010.

[76] Z. Qui, D. Yao, Y. Zhang, D. Ma, and X. Liu. The study of the detection of pedestrian and bicycle using image processing. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, 2003.

[77] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–187–I–194 Vol.1, June 2004.

[78] S. Ram, K. R. Ramakrishnan, P. K. Atrey, V. K. Singh, and M. S. Kankanhalli. A design methodology for selection and placement of sensors in multimedia surveillance systems. In *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks*, VSSN '06, pages 121–130, New York, NY, USA, 2006. ACM.

[79] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927 – 941, 2008. Semantic Knowledge in Robotics.

[80] I. Saleemi, K. Shafique, and M. Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *TPAMI*, 2009.

[81] S. Saravanakumar, A. Vadivel, and C. Saneem Ahmed. Multiple human object tracking using background subtraction and shadow removal techniques. In *Signal and Image Processing (ICSIP), 2010 International Conference on*, 2010.

[82] R. Satta. Appearance descriptors for person re-identification: a comprehensive review. *CoRR*, abs/1307.5748, 2013.

[83] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. People tracking with a mobile robot using sample-based joint probabilistic data association filters, 2003.

[84] W. Schwartz and L. Davis. Learning discriminative appearance-based models using partial least squares. In *Computer Graphics and Image Processing (SIBGRAPI), 2009 XXII Brazilian Symposium on*, 2009.

[85] S. Shahed Nejhum, J. Ho, and M.-H. Yang. Visual tracking with histograms and articulating blocks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008.

[86] P. Shirley and M. Ashikhmin. *Fundamentals of Computer Graphics, Second Edition*. Ak Peters Series. Peters, 2005.

[87] A. Smola and B. Scholkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[88] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999.

[89] K. Tarabanis, P. Allen, and R. Tsai. A survey of sensor planning in computer vision. *Robotics and Automation, IEEE Transactions on*, 11(1):86–104, Feb 1995.

[90] K. Tieu, G. Dalley, and W. Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1842–1849 Vol. 2, Oct 2005.

[91] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[92] D. Vasquez and T. Fraichard. Motion prediction for moving objects: a statistical approach. In *ICRA '04*, 2004.

[93] D. Vasquez, T. Fraichard, O. Aycard, and C. Laugier. Intentional motion on-line learning and prediction. *Mach. Vision Appl.*, Sept. 2008.

[94] J. Villalba Espinosa, J. Gonzlez Linares, J. Ramos Czar, and N. Guil Mata. Kernel-based object tracking using a simple fuzzy color histogram. In *Advances in Computational Intelligence*, Lecture Notes in Computer Science. 2011.

[95] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518 vol.1, 2001.

[96] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. *Proc. Computer Vision and Pattern Recognition, 2014.*, March 2014.

[97] Y. Wang, E. Teoh, and D. Shen. Lane detection using b-snake. In *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, 1999.

[98] J. Watada, Z. Musa, L. Jain, and J. Fulcher. Human tracking: A state-of-art survey. In *Knowledge-Based and Intelligent Information and Engineering Systems*, Lecture Notes in Computer Science. 2010.

[99] S.-K. Weng, C.-M. Kuo, and S.-K. Tu. Video object tracking using adaptive kalman filter. *J. Vis. Comun. Image Represent.*, 2006.

[100] M. Weser, D. Westhoff, M. Huser, and J. Zhang. Multimodal people tracking and trajectory prediction based on learned generalized motion patterns. In *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, 2006.

[101] X. Xu and B. Li. Rao-blackwellised particle filter for tracking with application in visual surveillance. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, 2005.

[102] K. Yabuta and H. Kitazawa. Optimum camera placement considering camera specification for security monitoring. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 2114–2117, May 2008.

[103] K. Yamaguchi, A. Berg, L. Ortiz, and T. Berg. Who are you with and where are you going? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011.

[104] Z. Ye and Z.-Q. Liu. Tracking human hand motion using genetic particle filter. In *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, 2006.

[105] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 2006.

[106] K. Yokoi. Probabilistic bprrc: Robust change detection against illumination changes and background movements. In *In IAPR Conference on Machine Vision Applications (MVA2009), number 5-1*, 2009.

[107] M. Yokoyama and T. Poggio. A contour-based moving object detection and tracking. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, 2005.

[108] R. Zhang and J. Ding. Object tracking and detecting based on adaptive background subtraction. *Procedia Engineering*, 2012. 2012 International Workshop on Information and Electronics Engineering.

[109] Y. Zhang, T. Lei, R. Barzilay, and T. Jaakkola. Greed is Good if Randomized: New Inference for Dependency Parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1013–1024, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[110] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[111] W. Zhou, H. Xiong, Y. Ge, J. Yu, H. Ozdemir, and K. Lee. Direction clustering for characterizing movement patterns. In *Information Reuse and Integration (IRI), 2010 IEEE International Conference on*, pages 165–170, Aug 2010.

[112] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. D. Bagnell, M. Hebert, A. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Proc. IROS 2009*.

[113] Z. Zivkovic and B. Krose. An em-like algorithm for color-histogram-based object tracking. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004.